

ROBINSON VIDA NORONHA

**FERRAMENTA DE AUTORIA E ENSINO DE PERÍCIAS
EM CIRCUITOS ELÉTRICOS**

Dissertação apresentada como requisito parcial
à obtenção do grau de Mestre, pelo Curso de
Pós-Graduação em Informática, Setor de
Ciências Exatas, Universidade Federal do
Paraná.

Orientador: Prof. Alexandre Ibrahim Direne

**CURITIBA
2001**

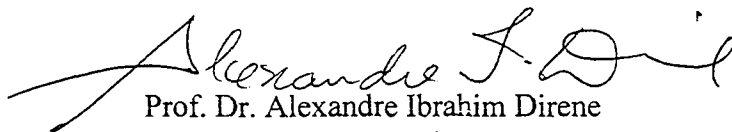


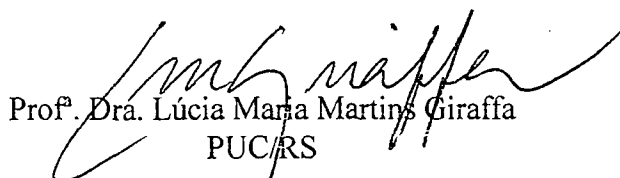
Ministério da Educação
Universidade Federal do Paraná
Mestrado em Informática

PARECER

Nós, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informática do aluno **Robinson Vida Noronha**, avaliamos o trabalho intitulado "**Ferramenta de Autoria e Ensino de Perícias em Circuitos Elétricos**", cuja defesa foi realizada no dia 16 de julho de 2001, às quatorze horas, na Sala de vídeo do Departamento de Informática da Universidade Federal do Paraná. Após a avaliação, decidimos pela aprovação do candidato.

Curitiba, 16 de julho de 2001.


Prof. Dr. Alexandre Ibrahim Direne
DINF/UFPR - Orientador


Prof.^a Dra. Lúcia Maria Martins Giraffa
PUC/RS


Prof.^a Dra. Aurora Trinidad Ramirez Pozo
DINF/UFPR

“Quero ensinar-te o que aprendi, mas isso eu quero dar gratuitamente, porque tu farás com esse aprendizado algo diferente do que fiz. E sei que, de algum modo, encontrarás a forma de me dizer que fizeste de forma diferente e porque o fizeste.”

R. Bach (Fugindo do Ninho)

Dedico este trabalho à minha família pelo carinho e apoio que sempre me deram, em especial ao meu irmão Alex, à minha namorada Ade pelo amor e companheirismo. Como não poderia deixar de faltar, ao meu amigo Caio Nakashima, por ter sido um grande incentivador nesta empreitada.

Ao meu amigo e professor orientador Alexandre Direne por ter conduzido este trabalho com tamanha maestria, sabendo corrigir os meus erros, mantendo-me no caminho do sucesso.

AGRADECIMENTOS

Agradeço ao Programa de Pós-Graduação em Informática da Universidade Federal do Paraná, em especial ao professor orientador Alexandre pela seleção. Agradeço também à professora Aurora, pelas contribuições feitas neste trabalho, aos professores Hélio Pedrini, Marcos Sunye, Olga, Laura e Michel Gagnon pelas excelentes aulas e amizade e ao Zé Carlos pela atenção e eficiência.

Aos meus colegas do curso de mestrado, em especial ao Eduardo Spinosa e Ernesto Rodrigues.

Aos meus amigos.

SUMÁRIO

LISTA DE ILUSTRAÇÕES.....	viii
LISTA DE TABELAS	ix
RESUMO	x
ABSTRACT	xi
1 INTRODUÇÃO	1
1.1 O PROBLEMA	1
1.2 VISÃO GERAL DO AMBIENTE AADDEE	2
1.3 ESTRUTURA DA DISSERTAÇÃO	4
2 HISTÓRICO.....	6
2.1 FERRAMENTAS DE AUTORIA E SISTEMAS TUTORES INTELIGENTES ..	6
2.2 SIMULADORES INTELIGENTES E NÃO INTELIGENTES	10
2.2.1 simuladores para circuitos elétricos	10
2.2.2 simuladores para outros domínios	12
2.3 ENSINO DE CONCEITOS INDUTIVOS	15
3 CONCEITOS E FERRAMENTAS DE ENSINO APRENDIZAGEM	19
3.1 CONCEITOS GENÉRICOS DE ENSINO E APRENDIZAGEM	20
3.1.1 elementos da aprendizagem.....	20
3.1.1.1 conhecimento de princípios sobre circuitos elétricos	21
3.1.1.2 conhecimento de perícia sobre circuitos elétricos	22
3.1.2 elementos de ensino.....	23
3.2 A FERRAMENTA DE ENSINO - FACE	27

3.2.1 Arquitetura Genérica Da Ferramenta.....	27
3.2.2 A Aplicação Cliente Da Ferramenta.....	28
3.2.2.1 A arquitetura	28
3.2.2.2 O módulo interface	29
3.2.2.3 O módulo codificador de respostas	30
3.2.2.4 O módulo decodificador de explicação	32
3.2.2.5 o módulo gerente de comunicações	33
3.2.3 A Aplicação Servidora	33
3.2.3.1 a arquitetura da aplicação servidora.....	34
3.2.3.2 gerente de comunicações servidor	34
3.2.3.3 simulador de circuitos elétricos.....	33
3.2.3.4 gerador de explicações – GEREX	34
4 CONCEITOS E FERRAMENTA PARA A AUTORIA DE EXERCÍCIOS.....	39
4.1 CONCEITOS E LINGUAGEM.....	40
4.1.1 Conceitos Básicos De Orientação A Objetos.....	40
4.1.1.1 Dado como foco da modelagem	41
4.1.1.2 Classe-instância	41
4.1.1.3 Reutilização de código por meio de herança	42
4.1.1.4 Definição evolutiva.....	43
4.1.2 Linguagem de Representação de Instâncias.....	44
4.1.2.1 Gabarito de simplificação.....	46
4.1.2.2 Gabarito de obtenção das equações de kirchoff.....	46
4.1.2.3 Gabarito de identificação de topologias	46
4.1.2.4 Gabarito de cálculo de valor de tensão e corrente	47
4.1.2.5 Gabarito de fornecimento de componente faltante	47
4.1.2.6 Gabarito de elaboração de circuitos por sinal de e/s	48
4.1.2.7 Gabarito de localizar defeito em circuito elétrico	48

4.2 FERRAMENTA DE AUTORIA – FAEX.....	49
4.2.1 Arquitetura Genérica Da Ferramenta Faex.....	49
4.2.2 A Aplicação Cliente Da Faex	50
4.2.2.1 O gerente de gabaritos	51
4.2.2.2 O gerente de topologias	52
4.2.2.3 Gerador de exercícios.....	52
4.2.2.4 Interface.....	53
4.2.2.5 Codificador de exercícios.....	54
4.2.2.6 Decodificador de exercícios.....	56
4.2.2.7 O módulo gerente de comunicações	57
4.2.3 A Aplicação Servidora	57
5 CONCEITOS E FERRAMENTA DE AUTORIA DE TOPOLOGIAS	59
5.1 CONCEITOS E LINGUAGEM.....	60
5.1.1 Conceitos Da Formação De Classes Em Orientação A Objetos	61
5.1.1.1 Modularidade	62
5.1.1.2 Abstração.....	62
5.1.2 Linguagem de Representação de Classes	63
5.1.2.1 Primitivas para a definição do vértice do grafo do circuito.....	64
5.1.2.2 Primitivas para a definição de ligações.....	64
5.2 A FERRAMENTA DE AUTORIA DE TOPOLOGIAS - FATO.....	64
5.2.1 Arquitetura Genérica Da Aplicação Cliente	65
5.2.1.1 Gerente de interface	66
5.2.1.2 Codificador de estruturas topológicas.....	68
5.2.1.3 Decodificador de estruturas topológicas	68
5.2.1.4 O módulo gerente de comunicações	68
5.2.2 Arquitetura Genérica da Aplicação Servidora.....	69
5.2.2.1 A arquitetura da aplicação servidora	69

5.2.2.2 Gerente de comunicações servidor	70
5.2.2.3 Arquivos de classes de topologias.....	70
5.2.3 A Interface de Trabalho do Professor-Autor	72
6 CONCLUSÃO E TRABALHOS FUTUROS.....	74
6.1 REAFIRMAÇÃO DA CONTRIBUIÇÃO	74
6.2 TRABALHOS FUTUROS.....	75
REFERÊNCIAS BIBLIOGRÁFICAS	77

LISTA DE ILUSTRAÇÕES

1 O AMBIENTE AADDEE.....	3
2 REPRESENTAÇÃO DE RESISTORES EM PARALELO.	26
3 ESBOÇO DA FERRAMENTA DE ENSINO -FEREN	27
4 ARQUITETURA DA APLICAÇÃO CLIENTE - FEREN	28
5 EXEMPLO DE INTERFACE PARA O MODELO DE EXERCÍCIO OBTENÇÃO DAS EQUAÇÕES DE KIRCHOFF	29
6 EXEMPLO DE INTERFACE PARA O MODELO DE EXERCÍCIO SIMPLIFICAÇÃO DE CIRCUITOS.....	30
7 REPRESENTAÇÃO GRÁFICA DE UM CIRCUITO ELÉTRICO	31
8 ARQUITETURA DA APLICAÇÃO SERVIDORA.....	34
9 REPRESENTAÇÃO GRÁFICA DE UM CIRCUITO ELÉTRICO.....	38
10 DIAGRAMA FUNCIONAL DA FERRAMENTA DE AUTORIA DE EXERCÍCIOS	50
11 DIAGRAMA DA APLICAÇÃO CLIENTE DE AUTORIA DE EXERCÍCIOS	50
12 EXEMPLO DO MENU INSTANTÂNEO DO COMPONENTE RESISTOR.....	53
13 REPRESENTAÇÃO GRÁFICA E EM FORMA DE GRAFO DA MESMA ESTRUTURA TOPOLÓGICA.....	63
14 DIAGRAMA EM BLOCOS DA APLICAÇÃO CLIENTE DA FERRAMENTA DE AUTORIA DE TOPOLOGIAS	65
15 ÍCONE QUE REPRESENTA A LIGAÇÃO ENTRE UM COMPONENTE E UM NÓ	67
16 ARQUITETURA DA APLICAÇÃO SERVIDORA DA FERRAMENTA FATO ...	70

17 EXEMPLO DE CÓDIGO REPRESENTATIVO DE ESTRUTURA TOPOLÓGICA ARMAZENADO NO SERVIDOR	71
18 INTERFACE DE FERRAMENTA DE AUTORIA DE TOPOLOGIAS.....	72

LISTA DE TABELAS

1 TABELA COM O CÓDIGO DA REPRESENTAÇÃO GRÁFICA DA FIGURA 3.4.....	29
2 EXEMPLO DE CODIFICAÇÃO DE UMA INSTÂNCIA DA CLASSE DE EXERCÍCIOS DE SIMPLIFICAÇÃO.....	55

RESUMO

Este trabalho descreve conceitos e ferramentas de software para a autoria e o ensino de perícias em solução de problemas sobre circuitos elétricos. Algumas das demandas do treinamento da perícia de um especialista em circuitos elétricos são discutidas, assim como as formas de automatizar parte deste treinamento. Nesta contribuição, além dos conceitos fundamentais sobre a modelagem do domínio de circuitos elétricos, também são definidos vários conceitos pedagógicos genéricos que se prestam para o desenvolvimento das habilidades periciais. A ferramenta de ensino é formada por uma interface cliente e uma aplicação servidora, a qual é composta de um gerador inteligente de explicações e um simulador de circuitos elétricos para apoiar o aprendiz na solução de problemas. A criação de material de ensino é feita por meio de duas ferramentas de autoria. A primeira delas permite ao professor elaborar topologias de circuitos elétricos genéricos, as quais representam classes de problemas. A segunda ferramenta de autoria é destinada à criação de exercícios específicos como instâncias das referidas classes de problemas.

ABSTRACT

This work describes concepts and software tools for the authorship and the teaching of expertise in solution of problems on electric circuits. Some of the demands of training of a specialist's expertise in electric circuits are discussed, as well as the forms of automating part of this training. In this contribution, besides the fundamental concepts on the modeling of the domain of electric circuits, there are also several defined generic pedagogic concepts that are rendered for the development of expert abilities. The teaching tool is formed by a client interface and a server application, which is composed of an intelligent generator of explanations and a simulator of electric circuits to support the apprentice in the solution of problems. The creation of teaching material is made through two authorship tools. The first allows the teacher to elaborate topologies of generic electric circuits, which represent classes of problems. The second authorship tool is used in the creation of specific exercises as instances of the referred classes of problems.

CAPÍTULO 1 – INTRODUÇÃO

Esta dissertação descreve conceitos e ferramentas de software para autoria e o ensino de perícias na solução de problemas sobre circuitos elétricos. Algumas das demandas do treinamento da perícia de um especialista em circuitos elétricos são discutidas, assim como as formas de automatizar parte deste treinamento.

1.1 O Problema.

O ensino de engenharia é ainda realizado como era feito há 50 anos atrás, pelo professor em frente ao quadro negro, com um pedaço de giz, apagador e apontador. Ou mais recentemente, utilizando-se de projetor para apresentar palavras, símbolos estáticos, fórmulas matemáticas e diagramas. A maioria das faculdades de engenharia conhecem pouco sobre como os estudantes aprendem conceitos, principalmente aqueles que são fundamentalmente práticos. As novas tecnologias educacionais oferecem a possibilidade de uma educação mais completa e mais interessante tanto na área de Engenharia, como nas demais áreas do conhecimento.

Das novas tecnologias disponíveis para o ensino de engenharia, o uso do computador em sala de aula, ou como ferramenta de auxílio ao professor, tem sido empregada graças a um constante barateamento dos equipamentos de informática e à sua grande capacidade de processamento, em grande parte, apenas de elementos numéricos.

O ensino de circuitos elétricos foi o enfoque deste trabalho como parte da perícia a ser adquirida nos estudos realizados em cursos engenharias onde Circuitos Elétricos é cadeira básica. O campo de ensino de circuitos elétricos

apoiado por computador tem se limitado apenas ao uso de simuladores ou ferramentas de análise matemática. Mesmo considerando a área de ferramentas inteligentes de ensino-aprendizagem, não há registro de linguagem de autoria e *shells* de ensino com aplicações nesta área de eletricidade. Murray (Murray, 1998; Murray, 1999) resume as últimas grandes pesquisas em autoria e *shells* inteligentes sem que algumas delas se prestem ao ensino de perícias em circuitos elétricos.

Ao mesmo tempo, não são conhecidas formalmente as capacidades isoladas da perícia de um técnico humano no reconhecimento de situações normais e detecção de falhas em circuitos elétricos. Com isso, torna-se difícil estruturar o ensino de tais circuitos já que esta é quase puramente uma tarefa de composições práticas.

1.2 Visão Geral do Ambiente ADDEE

O ambiente AADDEE (Ambiente de Autoria para a Descrição de Diretivas no Ensino de Eletricidade) proposto e implementado neste trabalho é composto de uma ferramenta genérica (*shell*) de ensino/aprendizagem (FACE - Ferramenta para Aprendizagem de Circuitos Elétricos) e de duas ferramentas também genéricas de autoria (FATO - Ferramenta de Autoria de Topologias e FAEX - Ferramenta de Autoria de Exercícios).

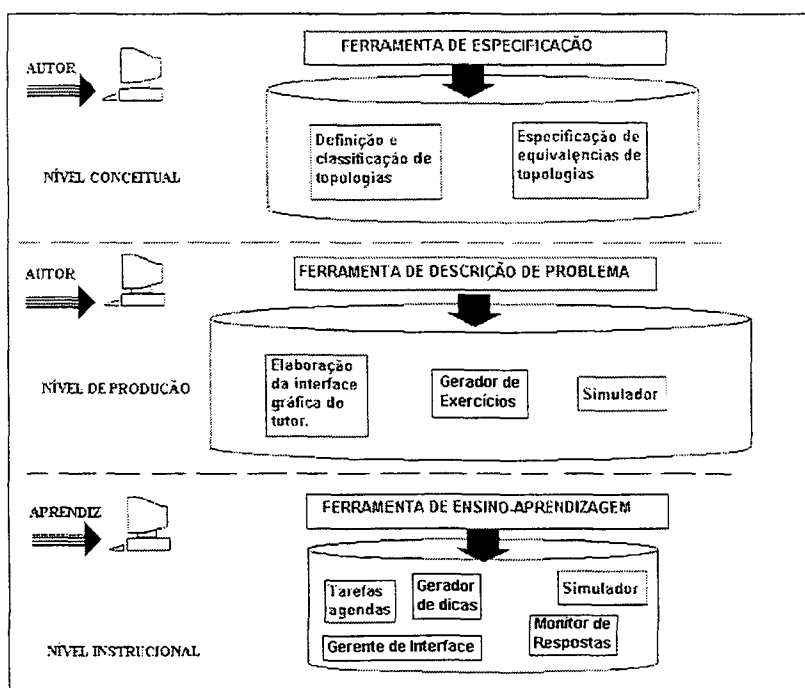
O ambiente foi concebido para funcionar em rede, onde existe a figura de um servidor. Junto com este servidor, está disponível ao autor e ao estudante, recursos de simulação e de acompanhamento inteligente da solução de exercícios. Do lado Cliente existe uma interface de manipulação direta onde professores-especialistas e aprendizes desenvolvem as suas tarefas.

A principal função do ambiente AADDEE é servir de auxílio para: a) o professor na produção de exercícios contendo problemas de circuitos elétricos; b) o aluno no monitoramento inteligente e interativo da solução dos problemas propostos.

O ambiente adotou um arcabouço genérico de autoria e interpretação pedagógica nos moldes do ambiente RUI (Direne, 1997). O ambiente RUI é destinado ao ensino de conceitos visuais. Uma arquitetura similar de representação e interpretação foi utilizada na ferramenta para o ensino e autoria de exercícios com circuitos elétricos.

A arquitetura funcional global do ambiente AADDEE é constituída de três módulos específicos, conforme ilustrado na Figura 1.1, sendo os dois primeiros (o Nível Conceitual e o Nível de Produção) voltados para o processo de autoria e o terceiro nível (O Nível Instrucional) voltado para o emprego de conceitos visando assim o desenvolvimento da perícia.

Figura 1.1 – O ambiente AADDEE



A ferramenta de especificação (autoria) de topologias FATO (Nível Conceitual), permite ao autor definir e classificar topologias de circuitos elétricos, equivalências simplificadas entre topologias, convenções a serem utilizadas pela ferramenta. Para a execução desta tarefa, o professor-autor conta com uma interface de manipulação direta de meta-elementos que devem ser utilizados de tal forma a representar uma estrutura topológica de um circuito elétrico real.

A ferramenta de autoria de exercícios FAEX (Nível Produção), possibilita que o professor elabore problemas de circuitos elétricos através da seleção dos procedimentos pedagógicos que estão disponíveis na ferramenta (*gabaritos*) e, a seleção da topologia de circuitos elétricos elaborada pelo autor na ferramenta de FATO.

A Ferramentas de Ensino/Aprendizagem (Nível Instrucional), permite que o estudante tenha acesso aos materiais instrucionais elaborados pelo professor com as duas ferramentas de autoria. A linha pedagógica disponível ao aprendiz é a de descoberta-guiada (*guided-discovery learning*).

Para o desenvolvimento das atividades neste ambiente, tanto o autor quanto o estudante contam com o auxílio de um simulador e um mecanismo de casamento de padrões de objetos desenvolvido especialmente para o ambiente.

1.3 Estrutura da Dissertação.

Este trabalho foi dividido em seis capítulos. No Capítulo 2 apresenta-se uma revisão de ferramentas e linguagens de autoria sendo genérica ou não, e ferramentas de simulação quantitativo ou qualitativo com e sem autoria. A

seção 2.3 é apresentado um resumo de trabalhos a respeito de ensino de conceitos indutivos.

O Capítulo 3 apresenta a ferramenta de Nível Instrucional. Esta ferramenta de interface dinâmica adapta-se às necessidades pedagógicas do exercício que é apresentado ao estudante. É realizado um acompanhamento das práticas do aprendiz e dicas podem ser fornecidas para obtenção da solução.

O Capítulo 4 descreve a ferramenta de nível de Produção. Esta ferramenta permite que o professor-autor instancie exercícios a partir de classes de atividades pedagógicas, e das topologias presentes no servidor. O autor ainda pode especificar algumas características gráficas tais como cor de fundo, fonte, entre outras.

O Capítulo 5 apresenta o processo de autoria de classes topológicas, além de fazer relações entre o processo de autoria e linguagens de orientação a objetos.

O Capítulo 6 apresenta a conclusão do trabalho, onde serão discutidas as contribuições deste trabalho e algumas linhas sobre trabalhos futuros.

2. HISTÓRICO

2.1 Ferramentas de Autoria e Sistemas Tutores Inteligentes.

Grandes diferenças podem ser apontadas entre os Sistemas Tutores Inteligentes - STI para as mais variadas áreas do conhecimento humano, construídos com ou sem ferramentas de autoria. Algumas das diferenças que podem ser destacadas são: (1) a ênfase na modelagem de domínio específico; (2) tarefas disponíveis para o processo de ensino aprendizagem; (3) profundidade com a qual os assuntos são abordados. De forma complementar, ainda podem ser citadas a metáfora de desenvolvimento empregada e o conhecimento de computação requerido pelo autor ou estudante para a correta utilização do STI ou da ferramenta de autoria.

Nos próximos parágrafos será apresentado um panorama de alguns trabalhos como representantes de Sistemas de Autoria e *Shells* de Ensino.

EON (Murray, 1998), possibilita ao autor construir a interface do sistema através de um grupo de objetos que contém propriedades tais como cor, tamanho e são capazes de armazenar instrução que respondem a eventos, como por exemplo o clique do mouse. Todas estas informações do desenvolvimento do domínio são armazenados em uma base de dados.

O sistema Eon fornece ao autor um modelo pedagógico fixo. A sua forma de programação gráfica baseada em uma linha de fluxo, permite que o autor, seqüencialmente, defina e ordene os procedimentos a serem executados pelo programa. Todavia o modelo pedagógico fixo pode ser interpretado por alguns autores como um limitador, pois o sistema produzido com esta ferramenta não possuirá a capacidade de gerar novos modelos que se adaptem ao perfil do usuário.

Blessing apresenta no DEMONSTR8 (Blessing, 1997) uma forma de programação por demonstração, onde o autor não necessita de conhecimento específico além do domínio a ser trabalhado, tornando assim o processo de autoria muito mais simples e rápido. O estilo de programação empregado por Blessing, utiliza o princípio de aprendizagem por repetição, onde o autor demonstra como resolver uma classe de problemas e o estudante pratica. Este estilo impossibilita o autor de definir estratégias pedagógicas específicas a serem empregadas em um determinado domínio. A ferramenta não dispõe de controle das ações do usuário nem permite a definição de estratégias para o ensino de determinado domínio. Este obstáculo de natureza pedagógica é capaz de comprometer a eficácia do sistema por empregar uma estratégia pedagógica desapropriada para determinado domínio.

REDEEM (Major, 1997) possui um conjunto fixo de procedimentos pedagógicos, mas o autor também pode definir e gerenciar a sua própria estratégia de ensino. Este poderoso recurso não é contemplado pelas ferramentas Eon e Demonstr8. O material de domínio no REDEEM é baseado na idéia de um catálogo de modelos de cursos. Esta ferramenta foi produzida em escala comercial como um pacote a ser trabalhado junto com a ferramenta ToolBook produzido pela empresa Asymetrix. Utiliza o paradigma de biblioteca de elementos, onde estes podem ser selecionados e utilizados no sistema. O autor é capaz de alterar e visualizar os valores dos parâmetros destes componentes, possibilitando assim uma adaptação do elemento de acordo com o domínio e os objetivos instrucionais. Esta visualização é feita através de um conjunto de slides e botões de seleção. Devido ao seu caráter de definição de estratégias, REDEEM tem sido apontado como sucessor do COCA (Major, 1991). Tutores no domínio da genética têm sido produzidos com o sistema REDEEM e aplicados no ensino de segundo grau (Murray, 1999). Esta ferramenta não permite a fácil manipulação matemática de expressões nem recursos para gerar simulação.

XAIDA (Redfield, 1996) apresenta ao estudante uma peça de um equipamento e é solicitado que identifique seus componentes, informe os passos de operação e de manutenção, ou aponte falhas e correções que podem ser feitas no equipamento. O tipo de abordagem utilizado pelo XAIDA caracteriza um modelo baseado em localização de componentes, instruções operacionais e uma grande quantidade de interações com o aprendiz. A maior dificuldade deste tipo de sistema é a definição da simulação dos componentes de um equipamento. O paradigma desta ferramenta é o do aprendizado pela execução. Recursos pedagógicos para transmitir instruções, conceitos e introduções são limitados pois esta ferramenta assume que o estudante possui um conhecimento básico do domínio que está sendo trabalhado.

RUI (Direne, 1997) é um sistema de autoria e ferramenta inteligente de ensino que contempla o ensino de conceitos visuais, aplicados a imagens médicas, onde o autor tem à sua disposição bons recursos para a elaboração de materiais pedagógicos. A metodologia pedagógica empregada é o aprendizado por descoberta guiada onde o estudante interage com o sistema através da inspeção de imagens e da tarefa de responder perguntas feitas pela ferramenta inteligente de ensino. Todavia, não existe nos recursos de autoria do sistema RUI elementos suficientes para a definição de equações matemáticas que representem o comportamento de objetos do mundo real. Isto impossibilita a representação precisa de conhecimentos de natureza procedimental, tal como o ensino de circuitos elétricos.

Sistemas tais como REDEEM (Major, 1997), EON (Murray, 1998) e COCA (Major, 1991) possuem um forte enfoque pedagógico, em detrimento da capacidade de definição de técnicas de solução de problemas em um determinado domínio. Apenas estratégias de ensino, por meio de linhas de fluxo por exemplo, fazem parte dos elementos de programação e autoria destes sistemas.

O Demonstr8 (aprendizado por demonstração) possui como característica principal a definição de regras e técnicas de solução de problemas no domínio. Ferramentas similares ao XAIDA (procedimentos de operação e componentes do equipamento), permitem que o autor trabalhe, de forma exploratória, as tarefas necessárias para o desenvolvimento de uma determinada perícia, onde nenhum recurso pedagógico adicional está disponível, exigindo assim do aprendiz um conhecimento básico do domínio que será trabalhado.

Como se pode observar, nenhum destes sistemas de autoria e “shells” de ensino atuais são capazes de combinar uma abordagem de discussão profunda sobre as características de um problema com um acompanhamento da solução do aluno para o referido problema. Nem mesmo os primeiros sistemas de autoria construídos contemplaram estas idéias.

Uma proposta de ferramenta, “CDT – Courseware Design Templates” (O'Shea et al, 1983), consiste de uma arquitetura genérica para Sistemas Tutores Inteligentes. Esta proposta nunca foi implementada, apesar dos grandes avanços no campo da linguagem de programação e ferramentas de autoria para STI – Sistemas Tutores Inteligentes. Dentre estes avanços, poderíamos destacar:

- Interface de manipulação direta,
- Mecanismos e ferramentas de complementação e interpolação simbólica de código,
- Checagem automática de aspectos de consistência e completude da base de conhecimento,
- Linguagens de alto nível de abstração e conhecimento,

- Ferramentas facilitadoras da reutilização de código.

Podemos assim, destacar uma carência de linguagens de programação e ferramentas de autoria para a criação de materiais pedagógicos que possam dispor de simuladores como uma das metodologias ou recurso.

2.2 Simuladores Inteligentes e não Inteligentes.

O estudo de circuitos elétricos ou eletrônicos, há vários anos, tem sido associado ao uso de simuladores como ferramentas de apoio no processo de aprendizagem. Professores e alunos utilizam o recurso de um simulador para validar a funcionalidade de um circuito. Um ambiente de aprendizagem baseado em simulação, possibilita que o estudante ensaie estruturas e descubra relações entre as mesmas; onde o aprendiz constrói o seu próprio conhecimento através da experimentação. Neste tipo de abordagem, um cuidado especial deve ser tomado pois, muitas vezes, o estudante não está apto à construção do seu conhecimento através da simulação (du Boulay e Mizoguchi, 1997). Fornecer uma estrutura para ser simulada, sendo ela um circuito elétrico ou o posicionamento dos planetas em torno do sol, envolve habilidades de geração de hipóteses, projeto do experimento, previsão e avaliação dos resultados; ou seja planejamento e monitoramento (du Boulay e Mizoguchi, 1997).

2.2.1 SIMULADORES PARA CIRCUITOS ELÉTRICOS.

Um dos simuladores de maior destaque, desenvolvido para o ensino de circuitos elétricos, é o SOPHIE (Brown e Burton, 1982) nas suas versões I,II e

III. Neste sistema, além de realizar simulações de circuitos elétricos, o estudante é capaz de interagir com o sistema, utilizando linguagem natural, fazendo ou respondendo perguntas. Também é possível interagir com o simulador, durante a execução do processo de simulação, alterando valores de parâmetros do circuito.

O SOPHIE I pode ser considerado como um ambiente reativo que não possui nenhuma iniciativa didática. A sua principal característica é a capacidade de responder questões hipotéticas, avaliar hipóteses formuladas pelo estudante e listar hipóteses possíveis de acordo com o circuito analisado. A segunda versão desta ferramenta, foi incrementada com mecanismos que possibilitam acompanhar as estratégias de resolução de falhas de circuitos elétricos. A última versão desta ferramenta foi projetada para ser um ambiente pedagogicamente ativo, centrado nas necessidades de aprendizagem do estudante.

O sistema SOPHIE nas suas versões I, II e III não possui ferramentas de autoria a ele acoplada, ou seja o professor não é capaz de alimentar o sistema com novos circuitos elétricos nem definir estratégias de ensino.

Assim como no SOPHIE, no QUEST (White et al., 1985), a representação interna de seus modelos utiliza-se de cálculos baseados nos componentes isolados de um circuito elétrico. O QUEST permite que estudantes testem as suas hipóteses fornecendo a descrição dos componentes de um circuito elétrico, solicitando ainda auxílio na solução do mesmo. Nesta ferramenta também não existe um módulo de autoria que possibilite ao professor o preparo de materiais de apoio ao ensino. O sistema é centrado no estudante e nas construções de seus protótipos. O sistema emprega as leis de Kirchoff para auxiliar à solução dos circuitos.

Mho (Lesgold, et al. 1987) é um ambiente de aprendizagem que dá apoio à livre exploração e à descoberta guiada. Este sistema gera as tarefas a serem executadas pelo estudante a partir dos tópicos a serem ensinados criando várias situações onde o referido domínio possa ser empregado.

Simuladores comerciais de circuitos elétricos podem ser utilizados como recurso pedagógico. Estes simuladores limitam-se apenas a resolver matematicamente um circuito fornecido. Não existe a autoria de cunho pedagógico e, muitas vezes, a simulação não condiz com o comportamento real do circuito. Este aspecto tem muitas vezes levado ao questionamento de conceitos teóricos por parte dos estudantes e professores. Observa-se também, uma discrepância nos valores apresentados em uma simulação dependendo do fabricante do simulador.

Estes simuladores não inteligentes limitam-se apenas a fornecer um valor de corrente elétrica e de diferença de potencial para um circuito elétrico. Nenhuma hipótese, fato teórico ou mesmo questionamento pode ser adicionado ao equacionamento matemático. Outra falha dos simuladores comerciais está na elaboração de circuitos elétricos sem a presença de resistores. Por exemplo, quando duas fontes de tensão são ligadas em paralelo, o simulador interpreta tal composição como um erro (Lyra, 1999), sem explicar ao aprendiz, o que tal situação pode representar no mundo real.

2.2.2 SIMULADORES PARA OUTROS DOMÍNIOS.

Já o simulador SIMQUEST (du Boulay e Mizoguchi, 1997) possui acoplado uma ferramenta de autoria. Neste sistema, através da manipulação direta de componentes previamente definidos, o professor instância objetos de uma biblioteca, criando assim um ambiente de ensino. O sistema possibilita ainda que perguntas sejam apresentadas ao estudante onde a sua resposta

afeta a simulação. O professor não possui recursos para elaborar exposição de conceitos novos aos estudantes, nem desenvolver ambientes que não possuam a característica da simulação. Durante a elaboração do ambiente de ensino, o professor define a equação matemática que será trabalhada na simulação, informação esta que não recebe sanção alguma, podendo, em alguns casos, conter falha. O estudante também não é capaz de definir modelos nem alterar a estrutura, mas somente fornecer valores para parâmetros e responder perguntas do sistema.

Um sistema de autoria produzido para a construção de simuladores inteligentes é o RECOVERY BOILER TUTOR (Woolf, 1987). Nesta ferramenta o estudante interage com o ambiente de uma caldeira, através de diversos recursos pedagógicos. Neste sistema o processo de autoria é através de programação nas linguagens C, Lisp e Fortran. Este aspecto de autoria requer um autor com conhecimento elevado em técnicas de programação.

Como ferramenta genérica, porém sem recurso de autoria, o SATELIT-1 (Direne, 1997b) possui a característica de simular a interface de um equipamento de controle de centrais telefônicas digitais. Nesta interface, o estudante fornece instruções textuais e o sistema simula o comportamento da central. Um elevado esforço foi feito no que concerne ao tratamento de erros sintáticos das instruções fornecidas pelo aprendiz. A singularidade desta ferramenta está nos aspectos genéricos de correção e explicação ("feedback") das falhas dos comandos digitados em sua interface. Isto é feito por meio de um algoritmo de busca heurística por melhor escolha ("best-first"). Através das explicações do sistema, o técnico ou engenheiro que está sendo treinado consegue adquirir a perícia necessária na programação da máquina. Em caso de mudança de treinamento de uma central telefônica específica para outra, basta que sejam fornecidas as descrições sintáticas da nova linguagem de

operação que o SATELIT-1 será capaz de ensinar está linguagem com base nos mesmos princípios pedagógicos.

Outro projeto sem capacidade de autoria mas com recursos de simulação, o STEAMER (Hollan et al, 1984) foi concebido para ser uma ferramenta instrucional para o treinamento de engenheiros que irão operar grandes navios. Neste ambiente o estudante trabalha com um cenário virtual de propulsão de um navio onde uma decomposição hierárquica permite que os estudantes explorem os subsistemas com maior detalhe.

O SMITHTOWN (Shute and Glasser, 1986; Shute and Bonar, 1986) é um sistema para ensino de economia onde o estudante é apresentado a uma cidade simulada, podendo variar parâmetros, tais como o preço de certas benfeitorias e estudar as modificações resultantes da simulação.

A partir do exposto acima, pode-se concluir então que há uma carência de simuladores com a finalidade não apenas de simular uma representação, mas sim que possua um comprometimento de ensino e desenvolvimento de perícia. Neste tipo de ferramenta o autor-professor prepara não somente um circuito a ser simulado, mas sim um domínio a ser ensinado e o estudante pode contar com uma análise dos resultados obtidos na simulação e interagir com o processo.

Até o momento, comparações sobre simuladores foram feitas apenas para produtos não inteligentes, não incluindo nenhum aspecto pedagógico. Sendo assim, parece ser primordial a valorização do autor para que ele possa impor sua capacidade pedagógica por meio de ferramentas de autoria aliada à capacidade de simulação.

2.3 Ensino de Conceitos Indutivos.

Conceitos fundamentalmente práticos podem ser interpretados como abstrações gerais derivadas de várias experiências. Um estudante pode abstrair pontos essenciais de uma leitura e escrever anotações a respeito da leitura (Howard, 1987). De forma similar, um estudante de engenharia através do domínio de princípios básicos de circuitos elétricos é capaz de descrever e compreender o funcionamento de todo aparato elétrico e eletrônico (Jackson, 1976).

O ensino de conceitos em alguns campos da Física e da Matemática, ao contrário da biologia, ciências sociais e humanas, necessitam de técnicas pedagógicas especiais (Howard, 1987). Estas técnicas incluem:

- exemplos e representação de protótipos;
- mapeamento de conceitos;
- uso de analogias e metáforas para definição de conceitos complexos.

Em geral, o ensino de tais conceitos práticos não se enquadra facilmente na teoria clássica de formação de conceitos. Isto ocorre pois a formação clássica de conceitos necessita de definições exatas, ou seja, deve possuir um número de propriedades necessárias e suficientes para descrever um objeto como instância de uma classe (Bussmann, 1991).

Na maioria dos casos de conceitos práticos, onde a definição exata de parte de um mundo não é possível, pode-se lançar mão da Teoria de Protótipos (Rosch, 1978). A teoria de protótipos não demanda uma definição exata, pois nem todas as instâncias de uma classe precisam ter os mesmos

atributos entre si. Adicionalmente, nenhuma instância de uma classe precisa possuir todos os atributos do protótipo da classe. A teoria de protótipos depende da noção de similaridade para ordenar e categorizar objetos como mais ou menos típicos de uma classe ou um conceito (Bussmann 1991).

A noção de similaridade é baseada na comparação de propriedades compartilhadas entre os objetos e as propriedades únicas de um objeto em relação ao grupo composto a partir do protótipo (Bussmann, 1991). Apesar de existir várias formas de concretizar esta idéia, a comparação anterior representa um ponto fundamental nas teorias de conhecimento e procedimentos sobre classificação (Tversky, 1977).

Merryl eTennyson, 1978, alerta para a necessidade de apresentação de exemplos e contra-exemplos na definição de conceitos. A estrutura desta apresentação (quantidade de exemplos e contra-exemplos, ordenamento, relacionamento entre exemplos e contra-exemplos, divergência e convergência dos exemplos e contra-exemplos) é condição essencial para o bom desempenho do estudante. Além disso, excelentes resultados foram obtidos com estudantes que, junto com exemplos e contra-exemplos, também receberam definições genéricas dos conceitos nos quais os exemplos se enquadravam (Tennyson e Park, 1980). Ainda em seu trabalho, Tennyson, alerta para o fato de estudantes que receberam apenas exemplos terem falhado com frequência na identificação correta de conceitos. Isto indica que a formação genérica de vários conceitos, sem contra-exemplos torna o aprendizado incompleto. Bussmann et al. 1991, em seu trabalho, implementou a representação de protótipos em um computador, respondendo a três principais questionamentos:

1. Como simular a formação do protótipo de um conceito ?
2. Que tipo de inferências serão possíveis ?

3. Quais informações devem ser armazenadas para a formação de conceitos ?

Para esta implementação, Bussmann utilizou a linguagem Common Lisp para a representação da formação de conceitos. A idéia principal é que o sistema receba a representação de um objeto como entrada e classifique o objeto utilizando uma estrutura conceitual que se desenvolve no processo. Grande parte das capacidades de uma perícia são desenvolvidas de forma indutiva, o que requer uma criteriosa organização da apresentação de exemplares para a boa formação de classes de conceitos por parte dos aprendizes. Todavia, o conjunto de tais exemplares e a ordem de apresentação do mesmos não conta com técnicas precisas na literatura (Sharples, 1991).

A literatura tem mostrado que a compreensão de conceitos e a capacidade de representação e classificação de protótipos em um determinado domínio científico são condições necessárias e suficientes para a formação de um perito. Lesgold (Lesgold, 1984) explora esta questão diferenciando as qualidades de um perito e de um novato. Por exemplo, para a solução de problemas de física, novatos tendem a decidir pelo emprego de equações rapidamente, selecionando aquela que inclui todos os dados do problema. Em contraste a este procedimento, especialistas inicialmente concentram-se em compreender o problema antes de aplicar equações.

Ainda em seu trabalho, Lesgold, (Lesgold, 1984) define que a estratégia de raciocínio progressivo ("working-forward") aplicada por especialistas requer as seguintes habilidades:

- Uma representação suficientemente robusta do problema que pode levar à sua solução;

- Conhecimento suficiente dos caminhos de inferências que podem ser seguidos para se atingir soluções parciais mais promissoras.

A interseção do trabalho de Lesgold, (Lesgold, 1984), com o de Bussmann, 1991, aponta que novatos possuem facilidades na aquisição de conhecimento quando a representação de um mundo é feita com a Teoria Clássica de Conceitos, ou seja, utilizando-se regras e procedimentos de classificação bem definidos. Por outro lado, ambos concordam que especialistas são capazes de empregar também a Teoria de Protótipos como base para a representação de seus conceitos mais avançados.

Uma proposta de uma arquitetura para a implementação de ambientes inteligentes de aprendizagem, o ELETROTUTOR (Silveira e Viccari, 97) nas suas versões II e III, trabalha com ensino de eletricidade, a lei de ohm e suas aplicações. Este projeto é estruturado por meio de lições, exemplos e exercícios, onde o estudante interage com a aplicação fornecendo dados sempre que solicitados e recebendo explicações. O ELETROTUTOR não possui a característica de ser uma ferramenta de autoria, permitindo apenas uma limitada capacidade de simulação centrada no exercício. A comprovação da eficácia desta ferramenta foi feita em sala de aula. Os alunos que utilizaram o ELETROTUTOR como material de apoio ao aprendizado apresentaram rendimento escolar superior aos estudantes que não tiveram contato com a ferramenta.

CAPÍTULO 3 - CONCEITOS E FERRAMENTAS DE ENSINO APRENDIZAGEM.

Dois grupos individuais compartilham da visão de que computadores podem prover excelentes ambientes de estudo a uma grande quantidade de estudantes.

O primeiro destes grupos, compostos basicamente por professores, conta com indivíduos experientes e pesquisadores na área de educação com grandes corpos de conhecimento específicos em seus domínios. Este grupo pode ser denominado de desenvolvedores de CAI (*Computer-Assisted Instruction*).

O segundo grupo é composto basicamente de pesquisadores da psicologia cognitiva e ciência da computação que se aperfeiçoaram no desenvolvimento de princípios e teorias de aprendizagem. Tais profissionais tentam empregar estas teorias tanto na concepção quanto no projeto de seus *softwares* educativos os quais possuem interfaces de caráter ativo. Este grupo pode ser denominado de desenvolvedores de STI (Sistemas Tutores Inteligentes).

Desenvolvedores de STI têm por objetivo implementar em programas de computador um conjunto de mecanismos suficientemente genéricos para fornecer ensino suplementar em uma variedade de aspectos pedagógicos. Estes programas tentam, de uma forma ou outra, através de bases de conhecimento representadas em lógica ou em outro formalismo, simular o comportamento de um tutor humano.

A ferramenta proposta neste trabalho tem por finalidade auxiliar no desenvolvimento de perícias em circuitos elétricos, fornecendo um conjunto de mecanismos genéricos que possibilitam ao aprendiz o correto emprego de

técnicas de análise de circuitos elétricos neste domínio. Neste ambiente, através da manipulação direta de elementos do circuito, teoremas matemáticos e técnicas de simplificação de maneira a solucionar problemas elaborados pelo professor.

3.1 Conceitos Genéricos de ensino e aprendizagem.

No ensino tradicional de circuitos elétricos, após terminar a discussão de um exemplo apresentado, o professor escolhe outro exemplo e continua a lição. Esta escolha, utiliza como base a experiência do professor, pois os próprios professores não sabem dizer exatamente quais os critérios usados para tal. Esta experiência somente é adquirida após muitos anos de prática do tutor, onde este, analisa a reação dos aprendizes e vai, aos poucos, descobrindo qual exemplo é o mais adequado a ser empregado.

O domínio dos conceitos é um componente de grande importância na perícia de resolução de problemas (Lesgold, 1984). O especialista em circuitos elétricos necessita dominar conceitos da prática pericial para que possa aplicá-los na solução dos circuitos ou na especificação de sua funcionalidade, bem como na predição de resultados.

3.1.1 Elementos da Aprendizagem

John Anderson (1982) através de sua teoria de aquisição de perícias, divide o processo em três estágios: (1) o da aquisição de conhecimento declarativo; (2) o da compilação de conhecimento; e (3) o da procedimentalização do mesmo conhecimento. Estas três divisões são empregadas, na maioria das vezes, pelos tutores em circuitos elétricos, onde

estes inicialmente explanam sobre os conceitos, apresentam estudos de casos e, a partir destes exemplos, o aprendiz é incentivado a exercitá-los em uma grande variedade de estruturas de circuitos elétricos. O presente trabalho enfoca a concepção de ferramentas de ensino que apóiam os estágios de compilação e de procedimentalização do conhecimento, segundo John Anderson (1982).

3.1.1.1 Conhecimento de Princípios sobre Circuitos Elétricos

A detenção do conhecimento de princípios é um pré-requisito para o desenvolvimento posterior de capacidades da perícia. Os elementos básicos que devem ser conhecidos pelo aprendiz em circuitos elétricos restringem-se a resistores, capacitores, indutores, fontes de tensão e corrente. Nos cursos de engenharia elétrica, estes elementos servem de arcabouço para o estudo de novos componentes tais como diodos, transistores, amplificadores operacionais e circuitos integrados em geral.

Cada um destes componentes possui um comportamento individual bem definido e formalizado pelas teorias ou princípios de circuitos elétricos. Estas teorias incluem desde princípios como a Lei de Ohm até métodos matemáticos que exigem do aluno a capacidade de manipular integrais, derivadas, transformadas de LaPlace, entre outras.

3.1.1.2 Conhecimento de Perícia sobre Circuitos Elétricos

Todavia, os componentes básicos de um circuito precisam sempre ser combinados (conectados) entre si para que uma finalidade esperada de comportamento do circuito elétrico seja alcançada. O especialista em circuitos

elétricos deve ser capaz de identificar características macroscópicas de um circuito, sendo a principal delas a topologia. A partir da identificação das características genéricas e particulares de um circuito, o que inclui a presença mínima de elementos em conjunto, um especialista pode: (a) discriminar entre topologias diferentes; (b) reconhecer topologias equivalentes; (c) calcular valores de um ou mais elementos a partir de outros; (d) completar partes do circuito; (e) reconhecer a forma de ligação entre elementos e (f) ser capaz de escrever, resolver e entender sem dificuldades as equações de circuitos. Estas são, de forma sucinta, as capacidades que compõem a perícia em circuitos elétricos.

Estas capacidades são desenvolvidas a partir de estudos de casos práticos, onde o tutor apresenta as técnicas necessárias e disponíveis para a solução de circuitos elétricos. De forma exaustiva, o aprendiz estende as referidas capacidades por meio da solução de problemas que incluem exemplos e contra-exemplos das classes topológicas fornecidos pelo tutor. O estudante, de forma indutiva, através de solução de problemas, entra em contato com uma grande variedade de estruturas ou topologias de circuitos.

A grande quantidade de topologias de circuitos a serem resolvidos segundo diversas técnicas, traz consigo, de forma empírica, o desenvolvimento da capacidade de selecionar qual a melhor técnica para resolver um problema, pois uma topologia que pode ser facilmente resolvida através de uma determinada forma, pode tornar-se um grande obstáculo se tentarmos por outra técnica. Esta particularidade é apontada por Howard em 1987.

Com relação ao reconhecimento de topologias equivalentes, pode-se dizer que esta capacidade depende da formação e reconhecimento de um conjunto de padrões de classificação. O perito, ao entrar em contato com um determinado circuito elétrico, consegue classificá-lo quanto à sua

funcionalidade, obter as funções matemáticas que descrevem o comportamento do circuito, e prever resultados que o circuito produz.

Já a capacidade de discriminar entre circuitos ou topologias diferentes é derivada do exercício de comparação entre diversos padrões de classificação conhecidos, ou seja, quando um circuito é apresentado a um especialista, este consegue apontar em quais classes este circuito não pertence.

3.1.2 Elementos de Ensino.

Com base no conjunto de capacidades da perícia em circuitos elétricos apresentadas acima, um instrutor pode planejar o ensino de cada uma das referidas capacidades de forma isolada. Para o planejamento deste ensino, algumas formas padronizadas de problemas podem ser propostas. Tais padrões de enunciados de problemas devem incluir elementos que exijam o completo entendimento da funcionalidade do circuito a partir dos elementos e suas ligações estruturais, bem como das técnicas de resolução a serem empregadas, fazendo com isso que o desenvolvimento destas capacidades ocorra.

No contexto deste trabalho, foram identificadas várias atividades de ensino que deveriam ser conduzidas para o desenvolvimento das capacidades da perícia. O elenco das principais atividades segue abaixo:

- a) Re-alocação de valores de componentes com correção por recálculo do simulador;
- b) Determinação de n elementos de um circuito a serem calculados a partir de valores de outros elementos;
- c) Representar o circuito a partir de um estímulo e uma resposta;

- d) Identificar qual elemento está faltando a um circuito e com que valor;
- e) Identificação de circuitos simples que sejam equivalentes, valorizando assim a autoria por meio dos parâmetros equivalentes de tensão, corrente, frequência, entre outras);
- f) Classificação de um circuito segundo sua topologia;
- g) Distinção de um circuito entre duas ou mais classes de topologias semelhantes.

No ensino de circuitos elétricos, princípios e perícias estão fortemente interligados, mas possuem fronteiras bem definidas. A distância entre estes dois tipos de conhecimentos descritos por Lesgold (Lesgold, 1984) pode ser percorrida através de estudo de casos e até mesmo de exercícios menos contextualizados.

Como exemplo, poderíamos citar: um dos conceitos do domínio de circuitos elétricos, o conceito de filtros de frequência, que é um circuito que atenua algumas frequências de um sinal pode ser representado por uma topologia composta apenas por resistores e capacitores, ou por resistores e indutores. Mas nem toda estrutura que possui estes componentes é um filtro.

O aprendiz de circuitos elétricos necessita saber interpretá-los para apontar as diferenças que o caracterizam e que o diferenciam da topologia de um filtro. Através da perícia de obtenção e interpretação das equações do circuito, este trabalho confirma os modelos de apresentação de exemplos e contra-exemplos de Merryl e Tennyson (1978) e Howard (1987) e Bussmann (1991).

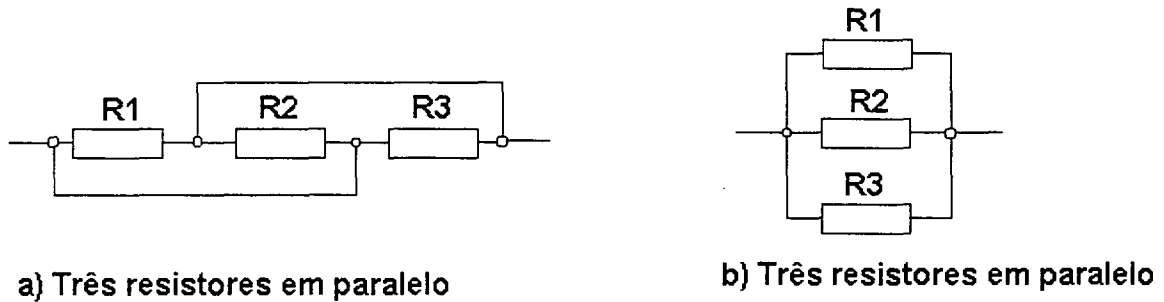
Na atividade de análise topológica de um circuito, o perito em circuitos elétricos inicialmente tenta lembrar-se desta estrutura. Caso seja uma estrutura nova e desconhecida, o segundo passo é identificar dentro desta estrutura, sub-estruturas conhecidas e a partir do conhecimento destas, de sua funcionalidade e equacionamento. Isto possibilita que o aprendiz possa se abstrair do comportamento total do circuito. As novas estruturas são analisadas a parte, ou seja separadas de seu contexto, e em seguida a sua funcionalidade e equacionamentos são analisados juntamente com o das estruturas conhecidas.

O aprendiz quando apresentado a um circuito, em muitos casos, não é capaz de analisar o circuito através de suas sub-estruturas, como faz o perito quando não conhece a topologia apresentada para um circuito muito grande.

No desenvolvimento da perícia de simplificação de um circuito elétrico resistivo, o aprendiz concentra-se nos componentes da estrutura, enquanto o perito concentra-se nas suas ligações, tentando identificar visualmente as conexões série e paralelo.

Na Figura 3.1 a seguir é representado nas letras a) e b) três resistores ligados em paralelo, ou seja, o mesmo circuito. Um especialista ao analisar ambas as estruturas, consegue identificar a equivalência mas um aprendiz não.

Figura 3.1 – Representação de resistores em paralelo.



Algumas das tarefas, atividades de ensino estudadas e formalizadas neste trabalho são:

- Apresentação de um número arbitrário (potencialmente grande) de exemplos concretos à luz de uma mesma classe de topologia;
- prática de solução de problemas de determinação de circuitos equivalentes;
- prática de obtenção das equações de kirchoff do circuito elétrico;
- prática de cálculo de valor de corrente elétrica e tensão em um determinado componente do circuito;
- prática de análise de funcionamento incorreto de circuitos elétricos ou eletrônicos.

Um principal destaque deve ser dado a circuitos compostos apenas por resistores e fontes de tensão. Todo e qualquer circuito elétrico ou eletrônico pode ser simplificado, ou reduzido, a elementos resistivos, e fontes de tensão

ou corrente. A finalidade da redução é facilitar o entendimento do comportamento do circuito e diminuir a quantidade e complexidade de cálculos necessários.

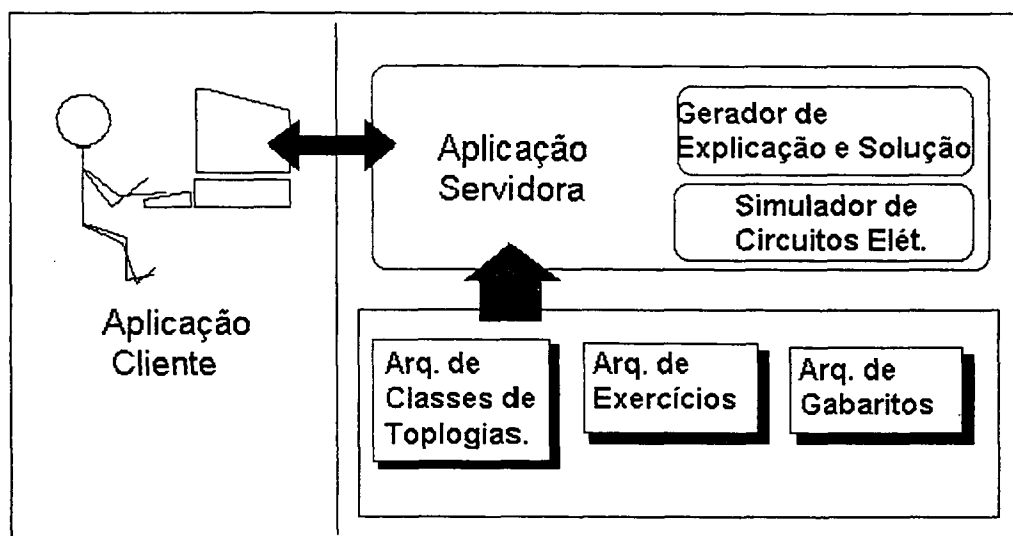
3.2 A Ferramenta de Ensino - FACE

O material de ensino elaborado pelo professor é apresentando ao estudante nesta ferramenta. Através de técnicas de descoberta guiada, o aprendiz tem a oportunidade de desenvolver algumas das perícias necessárias à manipulação, interpretação e análise de circuitos elétricos.

3.2.1 Arquitetura genérica da ferramenta

A Figura 3.2 apresenta, de forma resumida, a arquitetura da ferramenta de ensino. Por meio de sua interface, o estudante seleciona um exercício para ser objeto de estudo e monitoramento a partir da aplicação servidora.

Figura 3.2 – Esboço da Ferramenta de Ensino – FACE



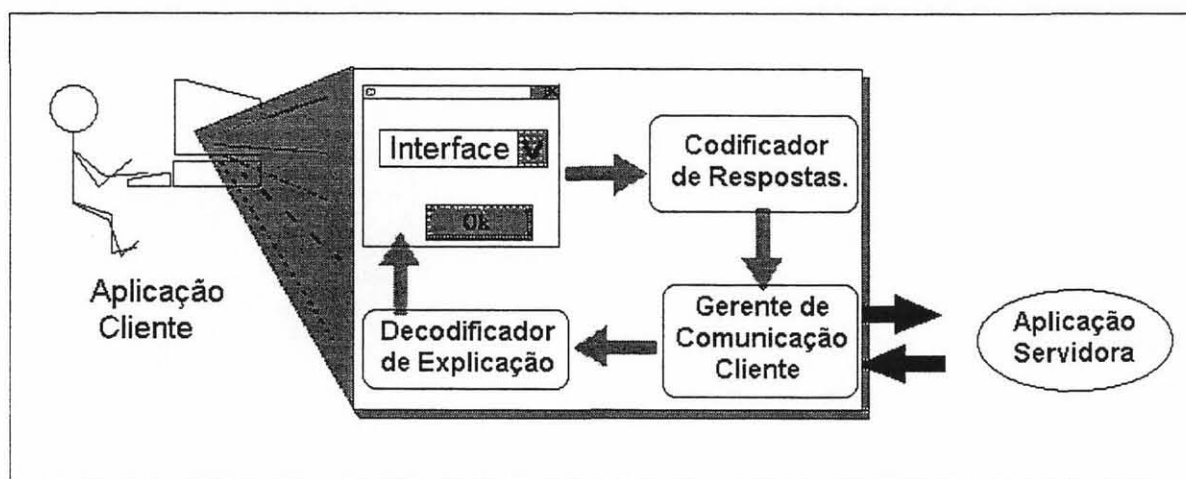
Durante a resolução do exercício selecionado, a aplicação cliente entra em contato com a aplicação servidora solicitando dicas, cálculos do simulador e informa as ações realizadas pelo estudante.

3.2.2 A Aplicação Cliente da ferramenta

3.2.2.1 Arquitetura.

De forma sucinta, as subseções seguintes apresentam a arquitetura da aplicação cliente, vide Figura 3.3. Ela tem, por principal função, servir de interface entre as ações do estudante e a aplicação “inteligente” servidora.

Figura 3.3 - Arquitetura da Aplicação Cliente - FACE



3.2.2.2 O módulo Interface.

O módulo Interface possui um interpretador de instruções, permitindo assim, a criação de elementos gráficos e de ações necessários à solução da questão. Estes componentes, ou elementos da interface, variam de acordo com a classe do exercício e a topologia do exercício. Esta variação abrange desde métodos e eventos a serem executados até tipos de componentes e seus valores. Estes elementos variam de funcionalidade, podendo ser botões que recebem cliques de mouse, caixas de listas de itens, elementos gráficos representando componentes elétricos, lista de eventos a serem executados, entre outros. As Figuras 3.4 a) e b) ilustram a dinâmica da interface que se adapta de acordo com o modelo do exercício.

Figura 3.4 a) Exemplo de Interface para o modelo de exercício
Obtenção das Equações de Kirchoff.

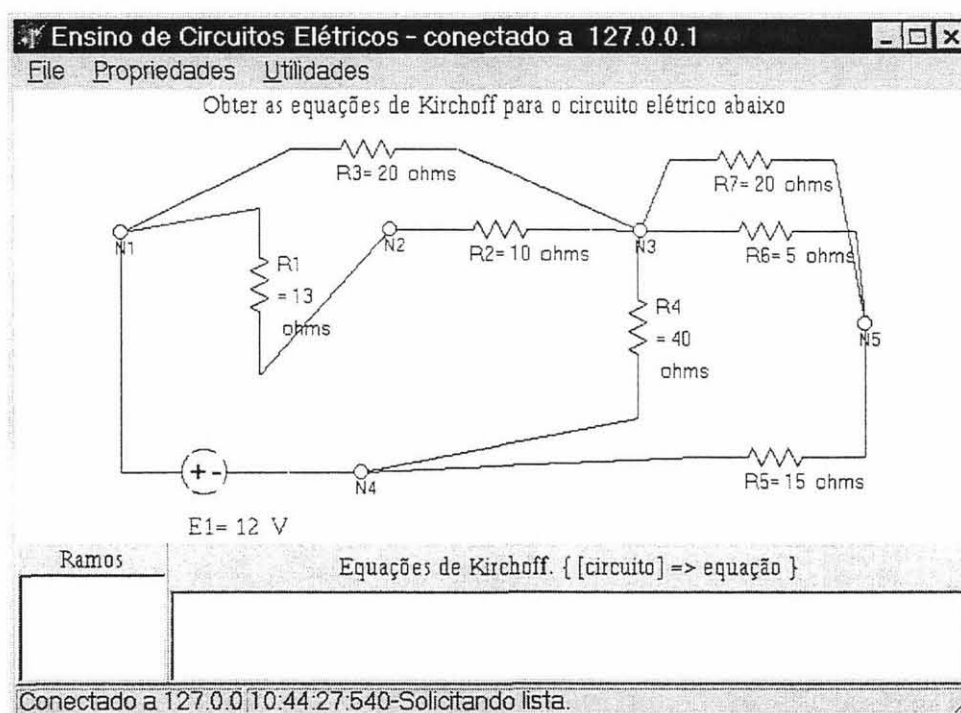
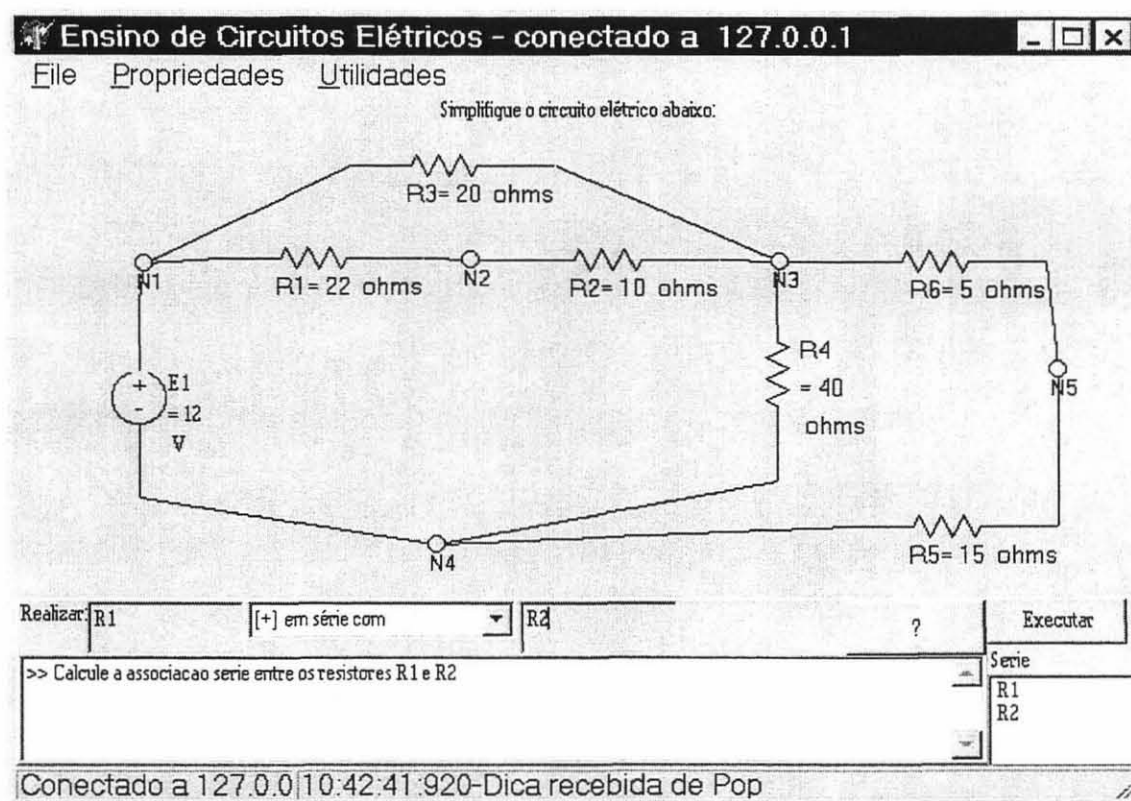


Figura 3.4 b) Exemplo de Interface para o modelo de exercício
Simplificação de Circuitos.



As únicas alterações que o estudante é capaz de realizar na interface são as ocasionadas em decorrência do processo da solução do exercício. Esta alteração, antes de ser processada, é encaminhada à aplicação servidora para uma validação.

3.2.2.3 O módulo Codificador de Respostas.

Este módulo permite a representação de forma textual das

características elétricas do circuito, bem como da requisição do aprendiz, tais como: ligações dos componentes elétrico, valores de resistências e tensões, executar a associação série entre dois elementos do circuito e calcular a tensão em um determinado resistor.

A codificação gerada tem dois objetivos: (1) enviar o circuito para o simulador (neste caso, é gerado o código no padrão *spice* do circuito – vide texto na tabela 3.1 com detalhes sobre o formato *spice*); (2) enviar o circuito para o gerador de explicações (o circuito é então traduzido para um formato de lista de itens aninhados com uma sintaxe determinada pelo Gerador de Explicações da Aplicação Servidora – vide texto na tabela 3.1). A decisão de qual formato escolher depende do “*gabarito*” utilizado no exercício, ou pode ser selecionado pelo próprio estudante como forma de auxílio na solução da questão.

Como exemplo, o circuito elétrico representado graficamente na Figura 3.5 é codificado conforme apresentado na Tabela 3.1 que se segue.

Figura 3.5 – Representação gráfica de um circuito elétrico

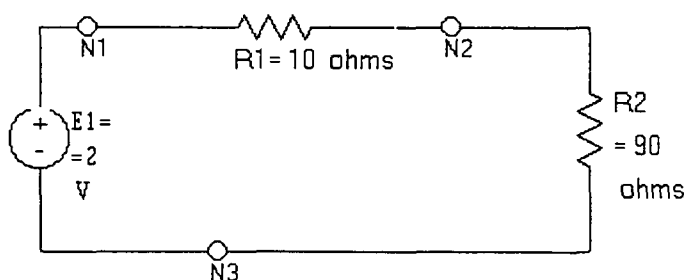


Tabela 3.1 – Tabela com o código da representação gráfica da Figura 3.5.

Código enviado ao Gerador de Explicações.	Código no padrão <i>SPICE</i> enviado ao simulador.
[v1 1 3 2.000e+00
[resistor [R1 [N1 N2] 10]]	R2 2 3 9.000e+01
[resistor [R2 [N2 N3] 90]]	R1 1 2 1.000e+01
[edc [E1 [N1 N3] 2]]	.controlc
] ->circuito;	op
	print all >>'...\simulador\saida.dat'
	.endc
	.end

3.2.2.4 O módulo Decodificador de Explicação.

As instruções enviadas pela Aplicação Servidora serão agora, interpretadas e apresentadas ao estudante. O resultado desta interpretação pode variar, podendo ser, por exemplo : (1) simples apresentação de dicas para a resolução do exercício; (2) resultados da simulação; (3) mensagens de erro caso a ação do estudante o afaste da resposta correta; (4) desenho ou

redesenho do circuito elétrico; (5) inserção ou retirada de ações da lista de ações.

3.2.2.5 O módulo Gerente de Comunicações.

O gerente de comunicações é responsável de forma direta pela interação entre a aplicação cliente e a servidora. Este módulo define quando enviar a mensagem e quando receber uma. Ele controla também a conexão podendo ativá-la ou encerrá-la. O protocolo de comunicação entre as aplicações é o TCP/IP.

3.2.3 A Aplicação Servidora.

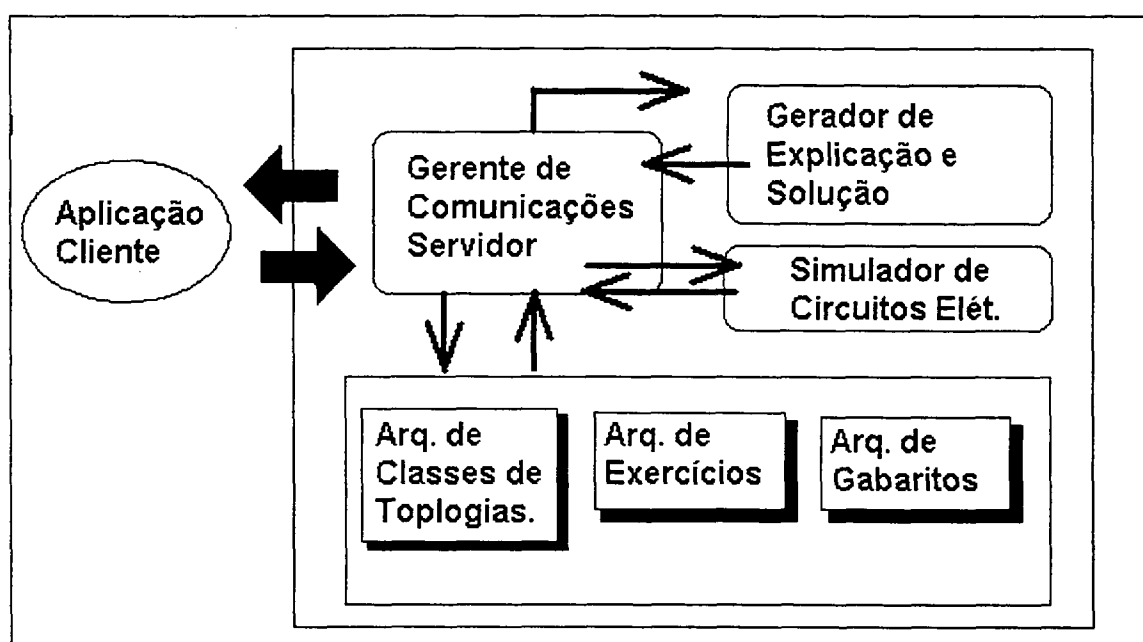
A aplicação servidora, que está localizada em um determinado ponto de uma rede TCP/IP, tem como função principal interagir exclusivamente com as aplicações clientes (nível instrucional, conceitual e de produção), fornecendo informações em resposta a determinadas solicitações. Para esta tarefa, a aplicação servidora conta com um simulador, um gerador de explicações e arquivos armazenados em disco.

Para a ferramenta de ensino, atualmente não há gravação permanente de dados na memória secundária. Isto é possível apenas para as ferramentas de autoria pois os exercícios definidos pelos autores devem estar disponíveis ao longo do tempo. Futuramente, esta ferramenta também efetuará gravação de dados para tornar permanente um modelo de estudante de longo-prazo.

3.2.3.1 A arquitetura da Aplicação Servidora.

A Aplicação Servidora tem por principal função fornecer uma interface entre as aplicações clientes e os arquivos armazenados em disco, gerador de explicações e o simulador de circuito elétrico. Sua arquitetura é representada de forma sucinta na Figura 3.6.

Figura 3.6 – Arquitetura da Aplicação Servidora.



3.2.3.2 Gerente de Comunicações Servidor.

O gerente de comunicações tem a finalidade de receber as solicitações das aplicações clientes, decidir qual das suas três principais funções básicas está sendo solicitada e realizar o que é pedido pela aplicação cliente. As três

principais funções são: (1) ler ou gravar dados no disco; (2) executar o gerador de explicações ou (3) executar o simulador de circuitos elétricos.

Além disto, deve retornar ao cliente o resultado da operação realizada, mantendo assim um diálogo com a referida aplicação.

3.2.3.3 Simulador de Circuitos Elétricos.

Um simulador de circuitos elétricos é uma aplicação que interpreta uma representação de um circuito, fornecendo como resultado desta interpretação valores de correntes elétricas ou tensões. O circuito elétrico fornecido ao simulador é equacionado em sistemas de matrizes que são resolvidas algebricamente segundo o método do Trapézio, Newton-Ramphson ou LU Decomposition.

Para o procedimento de simulação de circuitos elétricos, optou-se por utilizar o simulador aberto Winspice3 disponível no endereço "<http://www.willingham2.freemove.co.uk/winspice.html>". Este programa é produzido e distribuído na *University of California* pelo *Department of Electrical Engineering and Computer Science*, sobre a coordenação do professor Mike Smith.

Este simulador possui a característica de ser executado "em *background*", ou seja, qualquer aplicação pode solicitar a sua execução enviando o código a ser simulado (vide exemplo na Tabela 3.1). O simulador executa as instruções do código enviado e fornece um arquivo de saída com o resultado da simulação.

Além destas características, este simulador ainda possui uma pequena linguagem de programação que permite definição de variáveis, controles de

execução e decisão, bem como executar qualquer outra aplicação que estiver instalada no computador.

3.2.3.4 Gerador de Explicações – GEREX.

Tendo recebido a solicitação da aplicação cliente, o gerador de explicações compõe um fragmento de texto para promover a continuidade do diálogo reativo de respostas para a última entrada do aprendiz.

Ao receber a entrada do estudante, o Gerador de Explicações analisa o circuito sobre a ótica da perícia que está sendo trabalhada e verifica se o passo dado pelo aprendiz para solucionar a questão é um dos possíveis caminhos que poderiam ser empregados. Se o caminho não for válido, um pequeno texto corretivo é fornecido ao estudante. Caso contrário, o gerador de explicações não retorna este texto e sim um indicativo para a aplicação cliente processar o que foi solicitado pelo aprendiz.

O gerador de explicações GerEx, ainda é capaz de gerar todos os passos a serem executados pelo estudante, obtendo assim, a resposta da questão. Este recurso não está disponível na ferramenta de ensino, mas somente para uma das ferramentas de autoria.

O modelo de geração é genérico, conforme os princípios de interpretação pedagógica por meio de *shells* de ensino/aprendizagem. Tal modelo se baseia nas noções de: (1) violação do emprego de princípios sobre circuitos elétricos e (2) dicas de ações recuperadoras do caminho de solução.

Sendo assim, o GerEx se destina principalmente a comentar respostas erradas dos alunos. Isto, por si só, não soluciona diversos aspectos

pedagógicos de mais longo prazo, porém, serve como base para a criação de mecanismos reativos para o curto prazo.

Dos princípios que norteiam a geração de explicações, podem ser exemplificados:

- a) Associação em paralelo de dois componentes, se o estudante solicitar a associação em paralelo de dois componentes elétricos, por exemplo resistores, o gerador de explicações analisa o circuito e verifica se os referidos resistores podem ser associados desta forma.
- b) Seleção de ramos para a obtenção das equações de kirchoff, o aprendiz seleciona diversos ramos com o intuito de fornecer um caminho fechado para a corrente elétrica. A cada seleção, o gerador de explicações analisa se a inclusão deste novo trecho pertence a um caminho fechado e se não o leva a um caminho redundante.

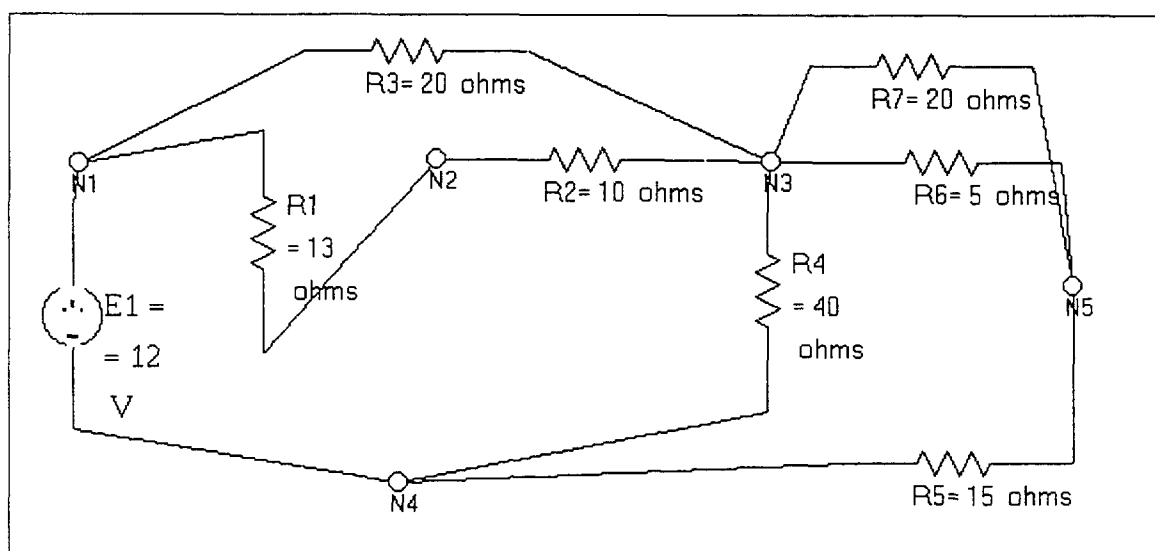
Além dos princípios, foi construída, para este módulo, uma máquina de casamento de padrões que correspondem à aplicação dos princípios. Por meio desta máquina, é possível comparar a entrada do aprendiz com cada princípio e, caso nenhum seja equivalente aos passos de aplicação do aprendiz, uma mensagem de erro é gerada. Juntamente com esta mensagem são apresentados os possíveis passos que poderiam ter sido tomados na solução.

Algumas das dicas de ações recuperadoras do caminho de solução são:

- a) No exemplo abaixo, o objetivo é obter as equações matemáticas que definem o funcionamento do circuito. O aprendiz seleciona o

ramo ou trecho do circuito através de cliques de mouse. No exemplo da mensagem, o aprendiz selecionou a fonte E1, o resistor R1 e o resistor R2. Os próximos componentes passíveis de seleção seriam : R3, R6, R7 ou R4 (vide Figura 3.7).

Figura 3.7 – Representação gráfica de um circuito elétrico.



Caso o aprendiz selecione o componente R3, o gerador de explicações informará ao estudante que:

“O componente R3 leva ao nó N1 que já foi visitado. Você deve tentar um outro componente.”

Na mesma topologia exemplificada acima, caso estivessemos trabalhando com o exercício de simplificação de circuito e através de cliques de mouse, o aprendiz solicitasse a associação em série dos resistores R5 e R7, o gerador forneceria uma mensagem corretiva informando que os referidos resistores não poderiam ser associados em série.

CAPÍTULO 4 - CONCEITOS E FERRAMENTA PARA A AUTORIA DE EXERCÍCIOS

O ensino de circuitos elétricos nos cursos de engenharia é realizado, muitas vezes, de forma intuitiva pelos professores. O instrutor da cadeira preocupa-se em apresentar os princípios básicos da eletricidade e fornecer uma grande variedade de exercícios aos aprendizes.

Os circuitos elétricos tipicamente apresentados são constituídos de elementos tais como resistores, capacitores, indutores e fontes. Estes componentes são interligados de tal forma que, a sobreposição de suas funcionalidades fornece um determinado comportamento a toda estrutura. Para justificar o comportamento de um circuito, o tutor baseia-se em equacionamentos matemáticos ou na experimentação.

Os princípios que são apresentados pelo tutor podem ser aplicados a uma grande variedade de circuitos, e um determinado circuito elétrico pode ser analisado segundo diversos princípios. Esta constatação, não tão óbvia ao instrutor com pouca experiência, indica que princípios e estruturas de circuitos elétricos são universos bem delimitados e que não possuem interseção. Durante a prática pedagógica, o tutor faz a interseção do circuito elétrico e do princípio que está sendo trabalhado, representando assim, uma determinada capacidade pericial.

Através da prática exaustiva de solução de exercícios, o aprendiz desenvolve a habilidade de selecionar o método mais adequado a ser empregado ao circuito. A Ferramenta de Autoria de Exercícios (**FAEX**) proposta neste trabalho possibilita a interseção de definições da estrutura

topológica de um circuito elétrico e dos princípios elementares da eletricidade. O resultado desta interseção é apresentado em um formato de exercícios ao estudante na ferramenta descrita no Capítulo 3.

4.1 Conceitos e Linguagem

Diante do que foi exposto no preâmbulo deste capítulo, como escolher uma forma de representação computacional de tal universo ?

Inicialmente, apelou-se para a necessidade de enquadramento da linguagem de autoria em um paradigma que auxilie o professor-autor a descrever e selecionar mecanismos pedagógicos que fossem apropriados ao processo de ensino/aprendizagem. Este processo deve se ater ao desenvolvimento da prática pericial que foi selecionada pelo tutor.

Este paradigma ainda deve ser capaz de facilitar a solução de problemas complexos de especificação, evitando a atividade de programação, e ainda podendo servir de macro-orientação na divisão do problema em subpartes consistentes e que permitam a reutilização. Sendo assim, o paradigma adotado para a criação da linguagem do presente ambiente de autoria é o da orientação a objetos. A conformidade das práticas pedagógicas com esta opção é apresentada nas subsecções abaixo.

4.1.1 Conceitos Básicos de Orientação a Objetos

Sistema baseado em objeto é aquele onde a computação é representada por uma série de entidades que interagem entre si para atingir um efeito desejado. Os princípios fundamentais da Orientação a Objetos utilizados no projeto da linguagem de autoria podem ser representados por: (1)

Dado como foco da modelagem; (2) Classe-instância; (3) Reutilização de código por meio de herança; (3) Definição evolutiva.

4.1.1.1 Dado como foco da modelagem

A visão tradicional do paradigma de programação imperativa é aquela onde comandos e procedimentos “atacam” os dados, sem uma grande preocupação com o estado de consistência resultante deste dado. Esta visão coloca em evidência a preocupação dos programadores com o comportamento e não com a integridade do dado, ao menos de forma primária.

No paradigma de programação Orientada e Objetos esta visão se inverte, o que pode permitir a criação de material integro, de qualidade razoável, e em maior quantidade para o ensino (Nicolson e Scott, 1986). Isto é atingido por meio de mecanismos de programação modular, os quais promovem grande *independência dos dados*, permitindo que sejam definidos de maneira auto-contida, incluindo seus aspectos de comportamento.

Neste trabalho, como a linguagem de autoria é uma categoria de linguagem de programação frequentemente destinada a (autores) leigos em Informática, o problema de integridade do material codificado se torna ainda mais crítico. Com isso, optou-se pelo paradigma de orientação a objetos para a criação da linguagem de autoria deste trabalho.

4.1.1.2 Classe-Instância

A base de composição genérica de uma classe de objetos como representação de conhecimento é a topologia de um circuito elétrico (vista em mais detalhes no Capítulo 5). A partir da definição de várias possíveis

topologias (ou classes), uma é selecionada pelo professor-autor para compor um exercício. Neste momento, a valoração específica (opcional) dos componentes integrantes da topologia genérica é feita por meio da instanciação desta estrutura topológica.

No processo de instanciação de uma classe para compor um exercício, a topologia do circuito elétrico assume o formato principal do estado da instância exercício. Isto está em conformidade com os preceitos já apresentados de independência entre topologias (classes) de circuitos e suas instâncias de exercícios. A definição de várias classes deverá cobrir uma grande quantidade de capacidades periciais que se deseja desenvolver no aprendiz. Além do mais, é possível enfocar uma das capacidades periciais específicas já apresentadas neste trabalho com a definição de apenas uma topologia e seus exercícios correlatos.

4.1.1.3 Reutilização de Código por meio de herança

Componentes elétricos, topologias de circuitos e gabaritos de exercícios são classificados como objetos neste ambiente de autoria, formando assim o vocabulário básico da linguagem de autoria que deve ser manipulado pelo autor. A manipulação de tais objetos é feita de forma gráfica, sem que o professor-autor necessite de treinamento específico em uma linguagem de programação.

Seguindo aspectos de Orientação a Objetos, a definição de um exercício como instância de uma classe topológica, por si só, já inclui a reutilização de dados íntegros por meio de herança de código. Além disto, também é possível reutilizar código de comportamento como sendo originados

de meta-classes de código muito genérico, as quais podem ser misturadas, por meio de herança, com as classes topológicas definidas pelo autor. Tais meta-classes contêm uma definição dos descritores de estados e métodos cuja essência é de natureza pedagógica.

A disponibilização do código de meta-classes por meio de gabaritos propicia a reutilização de comportamento pedagógico genérico (*padrão*) para toda e qualquer estrutura de exercício criado como instância de uma classe topológica. Assim como na reutilização de dados de circuitos, a reutilização de comportamento pedagógico é feita por meio de herança de código.

4.1.1.4 Definição evolutiva

O princípio de definição evolutiva admite mudança mínima (ou nula) em um código, acarretada como impacto decorrente da inclusão de novos fragmentos de código. No contexto deste trabalho, a definição evolutiva de novas classes de topologias causa impacto mínimo nas classes e exercícios existentes. Da mesma forma, a alteração de uma classe de topologia (exemplo, inclusão de um componente no circuito genérico) deveria causar alteração mínima ou nenhuma nos exercícios já criados como instâncias da mesma.

Na mesma linha, a definição e implementação futura de novos gabaritos de comportamento pedagógico podem ser incorporados à ferramenta de autoria e, conseqüentemente, sem necessidade de alteração das bases de ensino já existentes. A reprogramação da solução a ser apresentada pelo tutor por meio de um gabarito de comportamento pedagógico (já existente) também poderá mudar no código da meta-classe sem causar necessidade de alteração das topologias e exercícios criados.

4.1.2 Linguagem de Representação de Instâncias

A linguagem de autoria das ferramentas de software deste trabalho é orientada a objeto e suas primitivas foram criadas a partir da incorporação dos princípios de Orientação a Objetos citados em na Seção 4.1.1.

As primitivas permitem a criação de instâncias de classe de topologia (a definição de classes será explicada no Capítulo 5). Em tais instâncias, predomina: (1) a definição de valores concretos para elementos do circuito genérico que aparece na classe de topologia; (2) a complementação evolutiva do comportamento pedagógico padrão (gabarito) que um tutor deve possuir ao monitorar a solução de problemas.

Os gabaritos representam as primitivas de autoria e são de diversos tipos estando detalhados na próximas subseções. Foram criados 7 (sete) gabaritos de comportamento pedagógico, dos quais 2 foram implementados nos protótipo de ferramentas de software como primitivas de autoria para ilustrar o uso dos mesmos. Vale a pena lembrar que os sete gabaritos se originaram a partir da lista de itens de tarefas de ensino reativo apresentada a seção 3.1.

A ferrameta FAEX possui como principal função criar instâncias de exercícios a partir de gabaritos e classes de topologias.

Um objeto Exercício pode ser definido pelo autor de curso por meio do preenchimento de dados específicos para a seguinte estrutura genérica de dados e operações:

Estados:

Nome => nome da classe de exercícios.

Servidor => servidor que será utilizado : Wispice3 ou Poplog.

Interface => modelo de interatividade da interface.

Arquivo => arquivo a ser selecionado pelo servidor.

Função => função do arquivo que será executada.

Objetivo => objetivo do gabarito.

Componentes => descrição textual dos componentes que compõe o exercício, com seus respectivos estados e métodos.

Métodos:

Elaborar a Interface

Executar o Servidor

Ler o retorno do servidor

Apresentar mensagem do servidor

Executar instrução

Gerar descrição do circuito

Monitorar ação do estudante.

Os Estados e os métodos da classe variam de acordo com a classe de exercícios, podendo ter alguns acrescentados ou retirados.

4.1.2.1 Gabarito de simplificação (implementado).

Este gabarito desenvolve a capacidade do estudante para a simplificação de qualquer estrutura topológica de um circuito elétrico. Neste gabarito, o professor pode definir se o exercício trabalhará com valores numéricos, onde os elementos do circuito necessitarão de valores, ou se a questão será de caráter puramente literal, onde o exercício é apresentado de forma simbólica ao estudante.

Este gabarito trabalha com a habilidade de discriminar topologias, considerando neste caso as ligações em série e em paralela para o reconhecimento de topologias equivalentes, conforme descrito na seção 3.1.1.2.

4.1.2.2 Gabarito de obtenção das equações de kirchoff (implementado)

Através das equações de Kirchoff, qualquer topologia de circuito pode ter seu comportamento matematicamente definido. Este gabarito de exercício não trata da solução das referidas equações, mas apenas de sua obtenção. Esta característica está em conformidade com o que está apresentado na seção 3.1.1.2, onde destacamos a capacidade pericial por meio da qual um técnico é capaz de escrever e entender, precisamente, as equações de circuitos.

4.1.2.3 Gabarito de Identificação de Topologia

No ensino de circuitos elétricos, e principalmente em circuitos eletrônicos, algumas topologias clássicas são sempre apresentadas ao aprendiz. O conhecimento destas estruturas permite ao especialista a

identificação da funcionalidade de um circuito sem que seja necessário obter equacionamento algum. Nesta classe de exercício, há a necessidade apenas de valoração de componentes do circuito.

O valor dos componentes em uma estrutura pode definir o comportamento da estrutura. Assim, por exemplo, uma estrutura pode ser classificada como filtro *passa-faixa* ou filtro *rejeita-faixa* de acordo com os valores de resistores e capacitores. Nesta classe, a capacidade de reconhecimento de topologias equivalentes também é exercitada.

4.1.2.4 Gabarito de Cálculo de valor de tensão e corrente

Neste gabarito é exercitada a capacidade do aprendiz de prever os valores de corrente e tensão em um determinado componente do circuito elétrico. Neste gabarito, o recurso de simulação da ferramenta permite que professores valorizem os componentes de tal forma a não comprometer a finalidade do exercício. Ele permite ao aprendiz exercitar a perícia de cálculo de valores de um elemento, bem como facilita escrever e resolver, sem dificuldades, as equações de circuitos.

Este gabarito, ao ser executado pela ferramenta de ensino descrita no Capítulo 3 deste trabalho, não contará com o acesso ao simulador. Contará apenas com o Gerador de explicações.

4.1.2.5 Gabarito de fornecimento do componente faltante

Este gabarito é muito similar ao gabarito anterior, com a única diferença: o professor-autor irá selecionar o componente que estará oculto para o estudante na hora do ensino. Para compensar, o estudante tem acesso aos

valores de tensão e corrente aplicadas ao componente oculto. Discriminar entre topologias diferentes, reconhecer topologias equivalentes, reconhecer a forma de ligação entre elementos e ser capaz de escrever e entender sem dificuldades as equações de um circuito elétrico são as capacidades periciais destacadas na Seção 3.1.1.2 trabalhadas através da instanciação deste modelo de exercício.

4.1.2.6 Gabarito de elaboração de circuitos por sinal de E/S

Neste gabarito, o professor apenas necessita informar qual o componente será a Entrada e qual será a Saída. O gabarito, por si só, irá fornecer ao estudante todos os componentes da topologia que podem ser interligados e valorados, com exceção dos componentes responsáveis pela entrada e saída.

Durante a instanciação de exercícios deste gabarito o professor pode visualizar os valores de tensão na Entrada e na Saída. Estes valores são obtidos após uma consulta ao simulador. Perícias no cálculo de valores, completar partes de circuito, bem como a do reconhecimento de topologias equivalentes são enfocados nesta classe, conforme apresentado na seção 3.1.1.2.

4.1.2.7 Gabarito de localizar defeito em circuito elétrico

Uma das habilidades desejadas a um especialista em circuitos elétricos é a capacidade de prever o funcionamento correto ou incorreto de um elemento dentro de uma estrutura topológica.

O professor-autor deverá apenas selecionar o componente que estará com defeito e selecionar o defeito. Os defeitos desejáveis que o estudante tenha a capacidade de diagnosticar são: (1) Verificar se o componente está em curto-circuito (ausência de tensão em seus terminais); (2) Verificar se o componente está em circuito aberto (ausência de corrente elétrica); (3) Verificar se o componente apresenta o valor indicado (valores de corrente elétrica e tensão diferentes do esperado).

Sobre o enfoque da prática pedagógica, esta classe é que apresenta maior dificuldade para um aprendiz. Esta trabalha com todas as capacidades periciais descritas na seção 3.1.1.2.

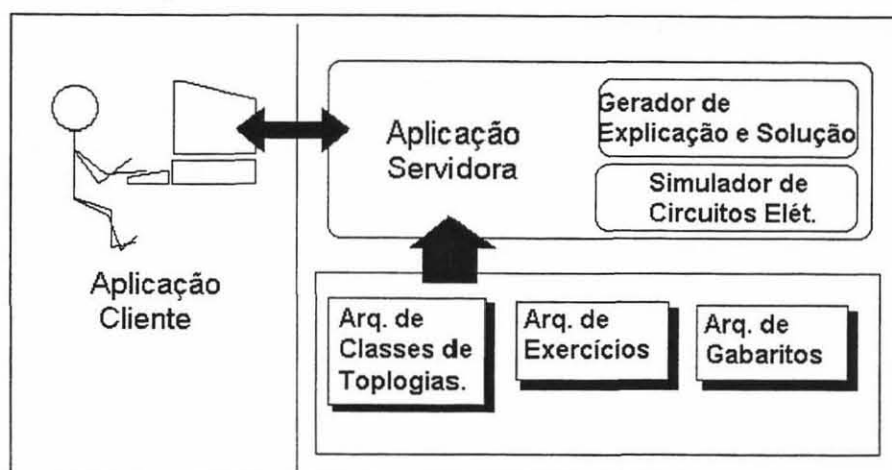
4.2 A Ferramenta de Autoria FAEX

Neste ambiente, o professor-autor elabora exercícios que serão apresentados aos estudantes conforme descrito no Capítulo 3. O paradigma da Orientação a Objetos deve ser aplicado aqui pelo autor, mesmo de forma intuitiva, na criação de exercícios por meio da instanciação de classes de topologias.

4.2.1 Arquitetura Genérica da Ferramenta FAEX

A Figura 4.1 apresenta, de forma resumida, a arquitetura funcional da ferramenta FAEX. Por meio da aplicação cliente, o professor-autor seleciona um meta-modelo de comportamento pedagógico por meio de um gabarito de exercício armazenado na aplicação servidora. Juntamente com uma classe de topologia selecionada, este gabarito será instanciado e o resultado será armazenado em formato de uma instância de exercício no servidor.

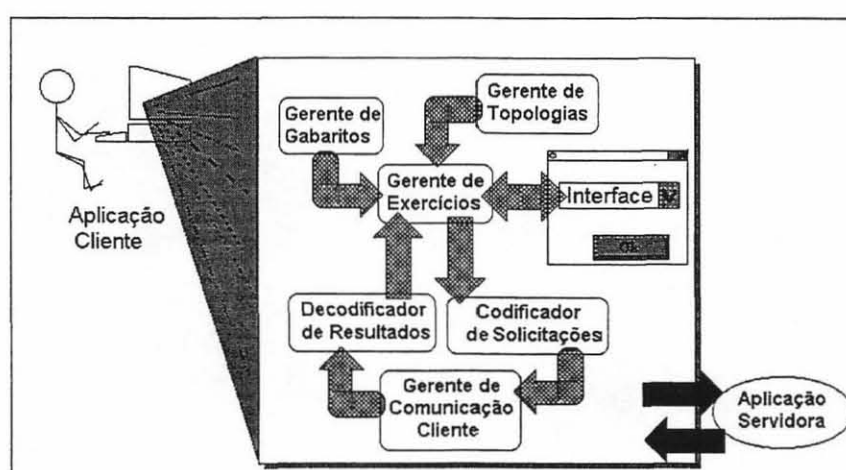
Figura 4.1 – Diagrama funcional da ferramenta de autoria de exercícios



4.2.2 A aplicação cliente da FAEX

De forma sucinta, as subseções seguintes apresentam a arquitetura da aplicação cliente representada na Figura 4.2. Ela tem, por principal função, servir de veículo para o professor-autor elaborar mecanismos computacionais específicos da prática de perícias em circuitos elétricos.

Figura 4.2 – Diagrama da aplicação cliente de autoria de exercícios.



4.2.2.1 O Gerente de Gabaritos

O gerente de gabaritos ou meta-classes de comportamento pedagógico, tem como principal função acessar e disponibilizar as informações necessárias a cada meta-classe diferente. O gabarito por si só não caracteriza um exercício a ser apresentado ao aprendiz. Ele possibilita sim, o melhor balizamento da definição de forma e de determinadas características que devem ser valoradas através do Instanciador de Exercícios. A meta-classe não possui nenhum conhecimento específico do tipo de topologia à qual o gabarito será associado. Isto oferece a vantagem de permitir que uma topologia seja definida e selecionada de forma independente da abordagem pedagógica responsável por ensiná-la, e vice-versa. A meta-classe de um gabarito armazena apenas as regras e os mecanismos pedagógicos das atividades (comportamento) a serem executadas pelo tutor e pelo aprendiz. Durante o processo de autoria, o autor deverá fazer um emprego coerente da classe de gabarito para a topologia associada. Ou seja, se o tutor-especialista selecionar o gabarito apresentado na seção 4.1.2.1, deverá selecionar também uma topologia que permita a condução de uma tarefa de solução de problema baseada na simplificação de circuitos.

Cada classe representada neste ambiente segue a teoria de protótipos de (Bussmann, 1991), onde nem todos os protótipos necessitam ter a mesma definição, nem estrutura. Esta flexibilidade possibilita uma independência de características entre conhecimento específico de domínio e conhecimento pedagógico. Em contrapartida, dificulta, mas não inviabiliza, a elaboração de mecanismos genéricos computacionais que sustentem permanentemente esta flexibilidade. Tal independência possibilitará que, no futuro, autores desenvolvam seus próprios gabaritos.

Para ilustrar o poder desta flexibilidade, um autor poderá empregar exemplos e contra-exemplos como mecanismo pedagógico básico (Merryl e

Tennyson, 1978). Isto será realizado através da criação e seleção do *gabarito* adequado ao desenvolvimento da perícia em foco.

4.2.2.2 O Gerente de topologias.

O gerente de topologias acessa e interpreta as informações de estruturas de circuitos elétricos transmitidas pela aplicação servidora. Cada topologia representa, de forma empírica, a dificuldade do exercício. Nela também são armazenados os seus componentes genéricos de circuito elétrico e suas ligações. Ao se projetar as topologias de maneira bem genérica, esta será a chave para atingir, com sucesso, futuras associações com gabaritos pedagógicos.

Nesta ferramenta, os componentes da topologia podem receber valor em termos de ohms, faradays, volts entre outros. Alteração na estrutura topologia não é permitida na FAEX, ficando esta tarefa confinada à ferramenta descrita no Capítulo 5.

4.2.2.3 Gerador de Exercícios

Nesta seção da aplicação, a instância do exercício é criada e apresentada ao autor, que deverá agora alterar algumas das suas características e propriedades. Tais elementos são: título da questão, cor de fundo, selecionar o componente que estará oculto (gabarito descrito em 4.1.2.5) e qualquer outra informação necessária a partir da classe de topologia.

O professor-autor poderá ainda lançar mão do simulador de circuitos elétricos da aplicação servidora e analisar a resposta do gerador de soluções. Todo exercício, após ser armazenado no servidor, pode ser editado ou

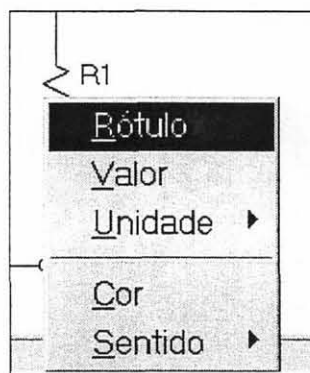
eliminado por esta ferramenta. A única alteração que não é possível é a troca de estrutura topológica.

4.2.2.4 Interface

Este módulo é responsável pela interpretação gráfica da instância de uma classe de topologia, bem como de preencher os estados da classe com as opções selecionadas pelo professor-autor.

Para a execução desta tarefa o professor possui ao seu dispor opções de manipulação direta por meio do emprego das técnicas de “clique” e de “arraste e solte”. Nesta versão da FAEX, o autor não possui mecanismos de inserção de códigos para alterar o comportamento pedagógico dos métodos da meta-classe de gabarito. Já os componentes da topologia possuem mecanismos de alteração de seus valores que podem ser acessados através do menu instantâneo que os acompanha, conforme ilustrado na Figura 4.3.

Figura 4.3 – Exemplo do menu instantâneo do componente Resistor.



Está previsto neste módulo, mas não implementado, a opção do autor inserir componentes não representativos de circuitos elétricos tais como caixas de edição, listas, imagens, entre outros.

4.2.2.5 Codificador de exercícios.

Este módulo permite a representação de forma textual das características do exercício, bem como das características elétrica do circuito, tais como: ligações dos componentes elétrico, valores de resistências e tensões, posicionamento na tela, entre outras.

A codificação gerada pode ter dois objetivos: (1) enviar a representação do circuito para o simulador (neste caso, é gerado o código no padrão *Spice* do circuito); (2) enviar o circuito para o gerador de solução (o circuito é então traduzido para um formato de lista de itens aninhados com uma sintaxe determinada pelo Gerador de Explicações e Soluções da Aplicação Servidora); (3) Gerar a representação textual do objeto exercício a ser armazenado no servidor (neste caso, optou-se por um formato bem simples que é ilustrado na Tabela 4.1). A decisão de qual formato escolher é feita pelo professor através de opções de menu ou da tarefa solicitada durante a autoria.

Como exemplo, a Tabela 4.1 apresenta uma parte de código da representação textual de um exercício que é gerado pela ferramenta e armazenado no servidor.

Tabela 4.1 – Exemplo de Codificação de uma instância da classe de exercícios de simplificação.

[EXERCICIO]

[TITULO] := simplificar o circuito.

[AUTOR] := Robinson Vida Noronha

[OBJETIVO] := Praticar o reconhecimento de ligações série e paralelo

Executar o cálculo série-paralelo para elementos resistivos.

[CABECALHO] := No circuito abaixo, calcular o valor da resistência equivalente.

[CORDEFUNDO] := 16777215

[CORDACANETA] := 0

[SERVIDOR] := TOPOP

[INTERFACE] := SIMPLIFIQUE

[FONT.NAME] := MS SANS SERIF

[FONT.SIZE] := 8

[FONT.COLOR] := -2147483640

[FUNCAO] := simplifique

[ARQUIVO] := simplifique.p

[DECLARE]

```
lbcabecalho : Tlabel

E1 : EDC

R1 : RESISTOR

.....

[BEGIN]

E1.left :=183

E1.top :=61

E1.orientation:= LEFTTORIGHT

E1.valor:= 12

E1.multiplo:=

E1.noA:= N1

E1.noB:= N2

.....

[END]
```

4.2.2.6 Decodificador de Exercícios.

Em situações de recuperação de dados criados anteriormente, as instruções enviadas pela Aplicação Servidora devem ser decodificadas e interpretadas para, só depois, serem apresentadas ao autor por meio da interface da FAEX. O resultado desta interpretação pode variar, podendo ser,

por exemplo : (1) a representação gráfica do exercício; (2) a apresentação dos resultados da simulação; (3) a apresentação da solução genérica gerada no servidor para a questão do exercício.

A representação gráfica do exercício consiste na criação e apresentação das classes indicadas na seção [DECLARE] do código (vide Tabela 4.1), bem como da modificação de suas propriedades declaradas na seção [BEGIN] ... [END].

4.2.2.7 O módulo Gerente de Comunicações.

O gerente de comunicações é responsável de forma direta pela interação entre a aplicação cliente e a servidora. Este módulo define quando enviar uma mensagem e quando receber uma. Ele controla também a conexão podendo ativá-la ou encerrá-la. O protocolo de comunicação entre as aplicações é o TCP/IP.

O gerente de comunicações é muito similar ao que foi apresentado na seção 3.2.2.6, sendo que a única diferença entre elas é a inclusão de caracteres de controle indicando qual ferramenta está solicitando serviço (neste caso, a FAEX).

4.2.3 A Aplicação Servidora.

A aplicação servidora, tem como função principal interagir exclusivamente com as aplicações clientes (nível instrucional, conceitual e de produção), fornecendo informações em resposta a determinadas solicitações. Para esta tarefa, a aplicação servidora conta com um simulador, um gerador de explicações e solução e arquivos armazenados em disco.

Para as ferramentas de autoria, há gravação permanente de dados na memória secundária. O arquivamento das instâncias de exercícios se faz necessária para a utilização destes arquivos na ferramenta de ensino descrita no Capítulo 3.

CAPÍTULO 5 - CONCEITOS E FERRAMENTA DE AUTORIA DE TOPOLOGIAS

Neste capítulo, são apresentados os conceitos e práticas necessárias e disponíveis ao professor-autor para a elaboração de topologias de circuitos elétricos. Tais conceitos, apesar de estarem apresentados apenas nesta parte final da dissertação, têm característica introdutória em relação à tarefa de autoria do professor. Neste ambiente, por meio da ferramenta de autoria FATO (Ferramenta de Autoria de Topologias), o professor irá elaborar o grafo genérico de uma topologia utilizando componentes (também genéricos) de circuitos elétricos.

Vale lembrar que instancias dinâmicas desta classe são criadas apenas com o auxílio da Ferramenta de Autoria de Exercícios (FAEX) descrita no Capítulo 4 e armazenada no formato de exercícios.

Sendo assim, neste trabalho, o processo de autoria foi desmembrado em dois estágios, auxiliados por duas ferramentas: uma para a autoria de exercícios (vide Capítulo 4) e uma para a autoria de classes de topologias (descrita neste capítulo). Esta separação tem dois motivos: (1) uma determinada topologia pode ser aplicada a diversas práticas pedagógicas, havendo assim a necessidade de uma evidente separação de classe de topologia e exercícios específicos; (2) a ferramenta de autoria de classes de topologia será cada vez menos utilizada pois a coleção (catálogo) de topologias deverá crescer e, ao se atingir uma quantidade suficiente, apenas a renovação permanente dos catálogos de exercícios será necessária como tarefa de autoria do professor-autor.

5.1 Conceitos e Linguagem

A linguagem apresentada para o desenvolvimento de ambientes reativos de ensino inclui características de orientação a objetos, contando assim com aspectos de modularidade, abstração, herança e poliformismo.

De forma antagônica às linguagens de programação do paradigma imperativo (tais como Pascal e C), este ambiente de autoria conta com conceitos de sintaxe e semântica do paradigma de orientação a objetos, conforme descrito parcialmente no Capítulo 4. A sintaxe da linguagem deste ambiente é definida pelos princípios básicos de análise de circuitos elétricos e métodos computacionais utilizados na solução de problemas sobre os circuitos.

As regras de sintaxe que o professor-autor deve considerar durante a elaboração das topologias são:

a) Circuito fechado - oriundo dos princípios de circuitos elétricos: toda e qualquer estrutura topológica de circuitos elétricos deve fornecer um “caminho fechado” para a corrente elétrica;

b) Terminal de um componente é ligado apenas a um Nó: um componente elétrico pode ser representado pelas quantidades de terminais que o mesmo possui (por exemplo, um terminal, dois terminais, três terminais e assim por diante - cada terminal do componente possui apenas uma única ligação, denominada de Nó);

c) Possibilidade de criar circuitos não resistivos¹: os simuladores de circuitos elétricos possuem a limitação de não permitir a interpretação de

¹ Denomina-se circuito não resistivo ao circuito que não possui o elemento resistor. Exemplo: duas fontes de tensão (baterias) ligadas em paralelo.

circuitos com ramos onde há ausência total de elementos resistivos - neste ambiente esta restrição não se aplica, porém, isto acarretará no fato de circuitos específicos não poderem ser simulados (pela ferramenta de ensino) quando derivarem de topologias desta categoria;

d) Possui um vocabulário limitado e não extensível - este vocabulário é composto apenas de Resistores, Fontes de Tensão e Corrente, Capacitores, Indutores, Terra, Nó, entre outros componentes elétricos.

A semântica da linguagem de autoria também é originada dos princípios de circuitos elétricos, onde cada componente é o modelo de um objeto no mundo real e consequentemente possui um comportamento único. Assim, todo e qualquer objeto que for representado como resistor, por exemplo, terá o mesmo comportamento e herdará as mesmas características de uma meta-classe Resistor, pré-definida na própria linguagem de autoria.

5.1.1 Conceitos da formação de classes em Orientação a Objetos

Além dos conceitos já apresentados na seção 4.1.1, alguns outros princípios de orientação a objetos foram aplicados na criação da linguagem de autoria de topologias. São eles os seguintes: (a) modularidade; e (b) abstração com auxílio de herança.

5.1.1.1 Modularidade

A unidade de modularidade em linguagens orientadas a objetos é a classe, a qual é composta pelo encapsulamento de dados e métodos, além de recursos de codificação abstrata. Cada classe é assim formada para que um programador (ou autor) possa *ocultar* informação, parcial ou totalmente, de acordo com seu objetivo. A ocultação de informação é importante pois alguns detalhes de implementação devem ser reservados ao criador da classe enquanto outros podem estar disponíveis para os usuários do código da classe.

Neste contexto, uma classe representa a estrutura topológica de um circuito abstrato que encapsula um conjunto de componentes e um conjunto de comportamentos pedagógicos (métodos). Destes elementos, apenas alguns detalhes estarão acessíveis aos aprendizes na fase de solução de problemas.

5.1.1.2 Abstração

A abstração está representada em sua estrutura, onde uma topologia pode ser composta por sub-estruturas que, na maioria das vezes, individualmente, possuem comportamento complementar ao geral. Estas sub-estruturas, por sua vez, são identificadas através de seu rótulo (muitas vezes um item gráfico). Além da descrição de uma classe em si já ser abstrata, a capacidade de abstração da programação como um todo pode ser elevada por meio da inclusão de *herança*. O recurso de herança, que é mais frequentemente aplicado no contexto de re-utilização de código, também tem uso nobre para a tarefa de abstração sobre dados.

Neste trabalho, a tarefa do autor de curso é altamente abstrata para a criação de topologias, efetuada quase sempre por meio da manipulação de rótulos gráficos. Tais rótulos são então utilizados durante o processo de autoria

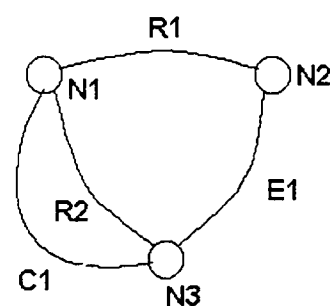
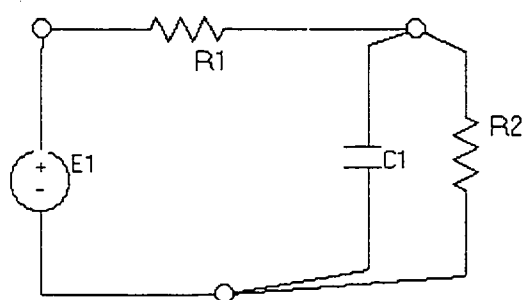
para abstrair os diversos atributos e métodos dos conceitos que representam o símbolo. O professor-autor tem, em seu ambiente de trabalho, um elevado nível comunicação com a ferramenta de autoria, se referindo aos meta-componentes de circuitos e aos algoritmos necessários para manipulá-los.

5.1.2 Linguagem de Representação de Classes

A linguagem de representação das classes é baseada na formação de um Grafo. Observamos assim que o uso de alguns meta-componentes pode, de acordo com as suas ligações no grafo, influenciar os diversos comportamentos de circuitos específicos.

Como exemplo do emprego da representação da estrutura de grafos, apresentamos a Figura 5.1-b.

Figura 5.1 – Representação Gráfica e em forma de Grafo da mesma estrutura Topológica



a) Representação de um circuito elétrico

b) Representação do Circuito Elétrico em forma de Grafo.

As Figuras 5.1- a) e b) possuem os mesmos componentes e as mesmas ligações, consequentemente representam a mesma estrutura topológica. A análise por grafo da estrutura topológica do circuito *não* constitui tarefa abstrata de autoria neste nível. Tal análise poderá ser feita, de forma ponderada ou não, pela aplicação de um certo gabarito pedagógico ao qual este grafo seria associado (vide Capítulos 3 e 4).

5.1.2.1 Primitivas para a definição dos vértices do Grafo do Circuito.

Os vértices do grafo de topologia representam os nós do circuito, os quais podem ser apresentados em qualquer ordem e percorridos em qualquer sentido. A partir deste vértice, é feita então a representação dos ramos com classes de componentes a partir de uma lista de meta-componentes disponível.

5.1.2.2 Primitivas para a definição das ligações.

As ligações são representadas pelas classes de componentes elétricos. A ligação, ou ramo, por si só, encapsula um conjunto de procedimentos e significados referentes à classe do componente. Este ramo, quando analisado juntamente com os demais, fornece o comportamento total, ou meta-classificação do circuito.

5.2 A Ferramenta de Autoria de Topologias - FATO

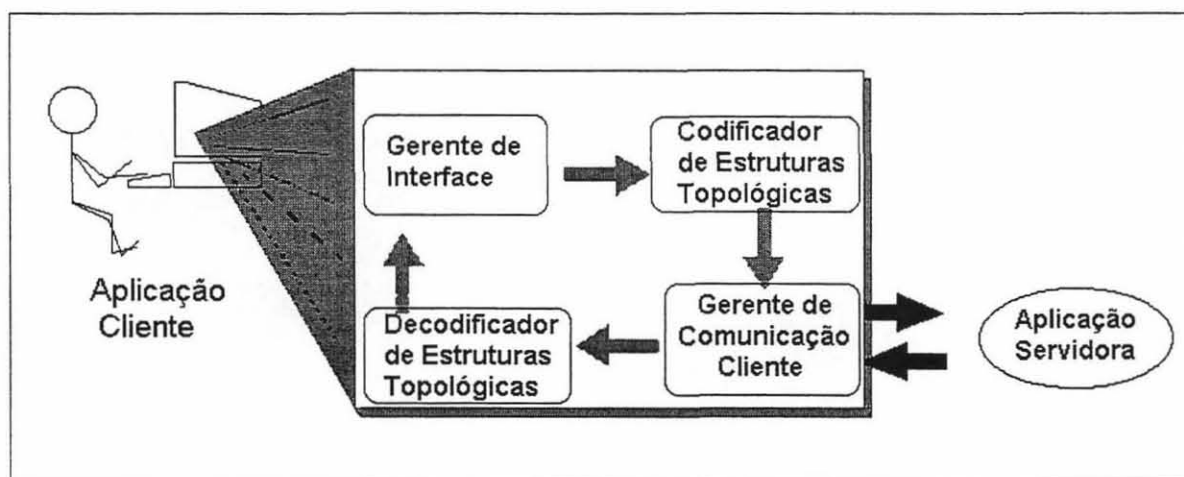
A ferramenta FATO é um veículo de autoria essencialmente visual, não fornecendo ao autor nenhum modo para fornecimento de instruções textuais.

As meta-classes que representam os componentes elétricos que estão disponíveis para o professor foram desenvolvidas para esta finalidade. O autor não tem recursos para modificá-las nem definir novas classes no protótipo que foi implementado até o momento.

5.2.1 Arquitetura Genérica da Aplicação Cliente.

A arquitetura genérica da ferramenta de autoria de topologias é apresentada na Figura 5.2.

Figura 5.2 – Diagrama em Blocos da Aplicação Cliente da Ferramenta de Autoria de Topologias.



A aplicação cliente é composta por 4 (quatro) grandes estruturas: Gerente de Interface, Codificados de Estruturas, Decodificador de Estruturas e Gerente de Comunicação cliente.

Nesta aplicação cliente, o professor-autor possui os mecanismos necessários e suficientes para a criação de estruturas genéricas que servirão de *gabaritos* nas ferramentas já descritas nos capítulos anteriores deste trabalho.

5.2.1.1 Gerente de Interface.

O Gerente de Interface, ou Gerente de Estruturas Topológicas fornece ao professor recursos de manipulação direta de meta-componentes na interface para que este “desenhe” a estrutura topológica do circuito. Como recursos para esta tarefa, o mesmo conta com meta-classes de componentes que são instanciadas através de técnicas de arrastar e soltar e cliques de mouse.

Estas meta-classes representam os componentes genéricos unitários da estrutura topológica, podendo ser: Resistores, Capacitores, Fontes de Tensão e de Corrente, Indutores, Terra e Nós.

Um componente da meta-classe Nó é quem gerencia a ligação entre os componentes. As ligações são armazenadas em uma lista. Este componente ainda é responsável pela representação visual em todo o ambiente de autoria das ligações entre os meta-elementos. A notificação da ligação a ser armazenada pelo componente é feita através de clique de mouse no ícone correspondente à ligação. Como exemplo de representação gráfica, a Figura 5.3 mostra a aparência do ícone de um componente da meta-classe Nó.

Figura 5.3 – Ícone que representa a ligação entre um componente e um Nó.



Durante o processo de autoria, o professor-autor conta com mecanismos de herança com encapsulamento dos elementos. Ao professor cabe a interpretação e classificação da estrutura topológica durante o processo de autoria. Não existe verificação alguma quanto à veracidade da classificação dada pelo autor. Esta característica valoriza o processo de autoria que só pode ser feita pelo professor-especialista em circuitos elétricos.

As características definidas na topologia serão herdadas pelas ferramentas já descritas neste trabalho através de criação de instâncias da topologia (exercícios). O autor não pode definir diretamente o comportamento da topologia (o comportamento da topologia é obtido em função do grafo de componentes).

A topologia gerada é genérica e não possui valoração dos componentes individuais. O único monitoramento de erro da ferramenta FATO para as tarefas do autor é dado em função das regras de sintaxe definidas em 5.1.

Uma topologia desenvolvida poderá ser utilizada como exemplo ou contra-exemplo de uma determinada classe (Tennyson e Merrill, 1978) na Ferramenta de Autoria de Exercícios descrita no Capítulo 4.

Cada classificação de uma estrutura topológica é armazenada em um arquivo diferente pela aplicação servidora. Isto vem de encontro à característica de poliformismo da ferramenta, permitindo assim que uma

determinada estrutura topológica possa receber várias classificações. A interface de acesso à topologia é bem definida, sendo esta restrita aos recursos incorporados na ferramenta.

5.2.1.2 Codificador de Estruturas Topológicas.

Este codificador tem como função principal armazenar em arquivo texto, todas as informações referentes à topologia da estrutura.

O arquivo gerado será transmitido através do Gerente de Comunicações para a aplicação servidora, sendo armazenado na memória secundária. Este registro é disponibilizado para a Ferramenta de Autoria de Exercícios. O arquivo pode ser editado ou mesmo retirado do servidor sem afetar os exercícios que foram instanciados com esta estrutura conforme apresentado no Capítulo 4.

5.2.1.3 Decodificador de Estruturas Topológicas.

O decodificador tem como função interpretar de forma visual as informações armazenadas na aplicação servidora. Esta interpretação conta com o auxílio do gerente de interface para representação visual dos componentes e suas ligações.

5.2.1.4 O módulo Gerente de Comunicações.

O gerente de comunicações é responsável de forma direta pela interação entre a aplicação cliente e a servidora. Este módulo define quando

enviar a mensagem e quando receber uma. Ele controla também a conexão podendo ativá-la ou encerrá-la.

Diferente do Gerente de Comunicações apresentado nos Capítulos 3 e 4, este não necessita estar ativado para que o processo de autoria ocorra. A ligação com o servidor é necessária apenas no instante da gravação ou leitura de estruturas topológicas. Toda a autoria pode ser realizada com o Gerente inativo.

5.2.2 Arquitetura Genérica da Aplicação Servidora.

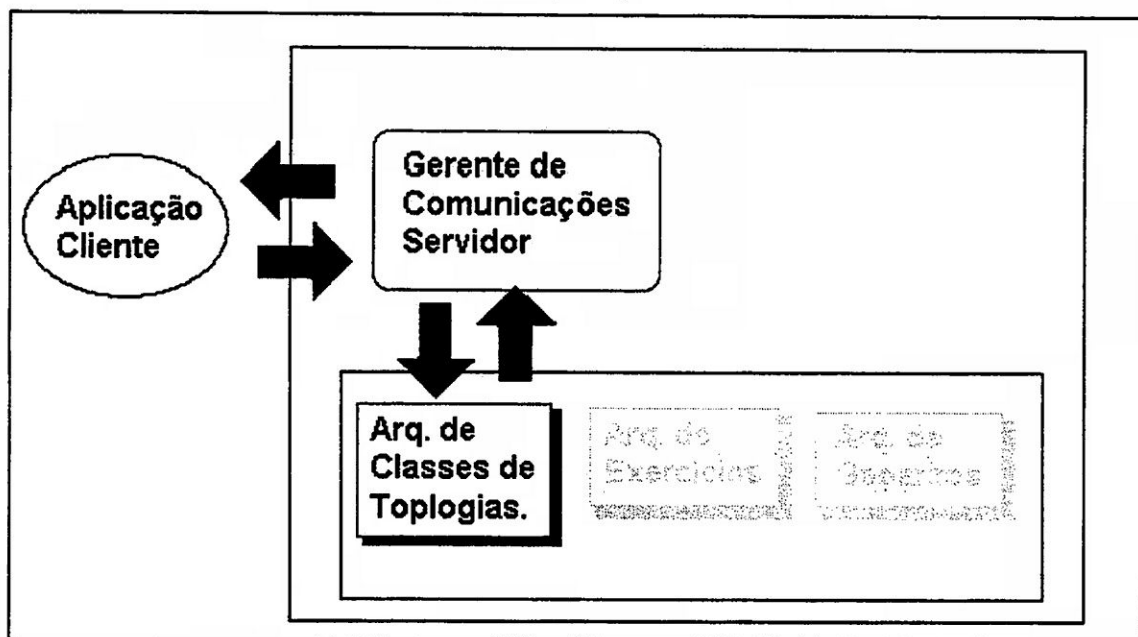
A aplicação servidora, que está localizada em um determinado ponto de uma rede TCP/IP, tem como função principal interagir exclusivamente com as aplicações clientes (nível instrucional, conceitual e de produção), fornecendo informações em resposta a determinadas solicitações.

Para as ferramentas de autoria (FATO e FAEX), há gravação permanente de dados na memória secundária. A gravação é feita independentemente da identidade do autor, não havendo controle algum quanto a direitos de escrita e gravação por parte de diversos professores que venham utilizar esta ferramenta. Futuramente, esta ferramenta efetuará este controle.

5.2.2.1 A arquitetura da Aplicação Servidora.

A Aplicação Servidora tem por função principal ser uma interface entre as aplicações clientes e os arquivos armazenados em disco, o gerador de explicações e o simulador de circuito elétrico. Sua arquitetura é representada de forma sucinta na Figura 5.4.

Figura 5.4 – Arquitetura da Aplicação Servidora da Ferramenta FATO.



5.2.2.2 Gerente de Comunicações Servidor.

O gerente de comunicações tem a finalidade de receber as solicitações das aplicações clientes para a ferramenta FATO. O gerente de comunicações possui apenas acesso de escrita e leitura nos arquivos de topologias.

5.2.2.3 Arquivos de Classes de Topologias.

Os arquivos armazenados no servidor possuem um formato de representação muito similar ao já apresentado neste trabalho. O arquivo é composto de duas seções, a primeira descreve os meta-elementos que pertencem à classe e, em segundo, a descrição de informações básicas para o

redesenho da estrutura topológica. A Figura 5.5 apresenta um trecho do código referente à estrutura topológica elaborada pelo professor-autor.

É importante notar que em nenhum instante se apresenta a classificação explícita da estrutura topológica. Uma descrição de classe só é efetivamente composta no instante de salvamento da topologia, ocasião em que o autor também fornece um título para a mesma.

Figura 5.5 – Exemplo de código representativo de estrutura topológica armazenado no servidor.

```
[DECLARE]
R1 : RESISTOR
.....
n2 : NO
[BEGIN]
R1.left := 173
R1.top := 61
R1.orientation:= LEFTTORIGHT
R1.noA:= n1
R1.noB:= n2
.....
n1.left := 113
n1.top := 74
n2.left := 301
n2.top := 75
n3.left := 189
n3.top := 198
[END]
```

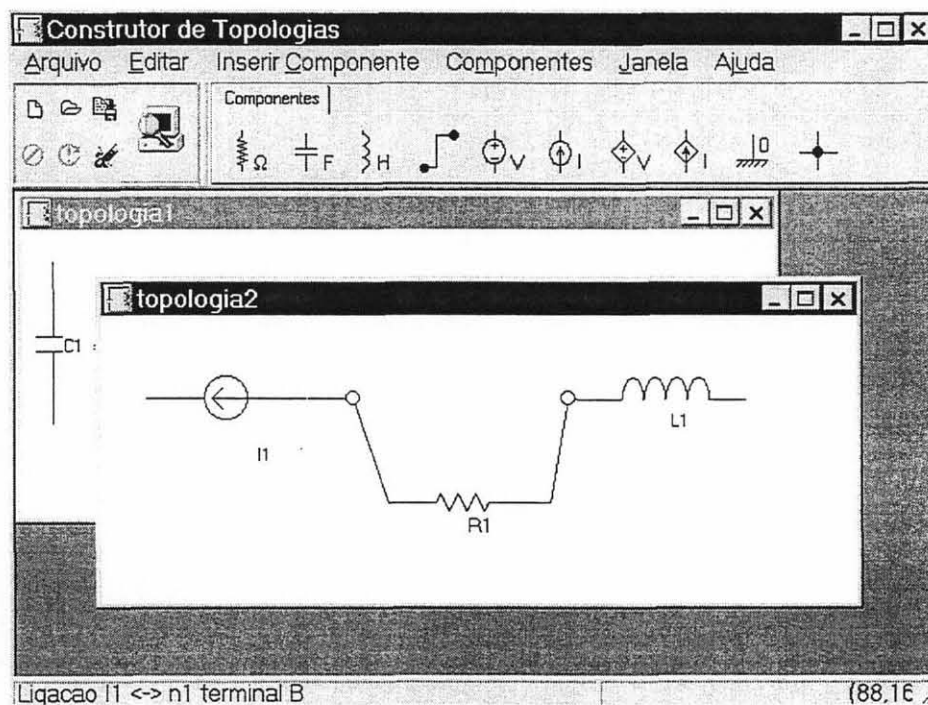
O professor-especialista, ao analisar a representação gráfica correspondente ao código, consegue e abstrair as possíveis classificações da instâncias da topologia. As vantagens da aplicação do conceito de classe-instância torna-se evidente por ocasião do uso da Ferramenta de Autoria de Exercícios (FAEX).

5.2.3 A Interface de Trabalho do Professor-Autor.

A interface de autoria, que permite a elaboração de classes de topologias de circuitos, é apresentada na Figura 5.6

Foi uma preocupação deste trabalho elaborar uma interface com características de um ambiente visual de programação.

Figura 5.6 – Interface de Ferramenta de Autoria de Topologias.



A ferramenta é composta de uma única interface, que engloba os quatro módulos descritos anteriormente nas seções 5.2.1.1 a 5.2.1.4.

Neste ambiente não existe interface para a valoração dos componentes, procedimento este realizado pela ferramenta descrita no Capítulo 4. O autor possui ao seu dispor apenas os meta-componentes apresentados na Paleta de Componentes. Futuramente esta paleta será expandida para disponibilizar componentes eletrônicos tais como Diodos, Transistores e amplificadores Operacionais. Para os professores que adquirirem perícia neste ambiente, existe a possibilidade da interface ler de um arquivo texto externo, informações topológicas apresentadas com a sintaxe exemplificada na Figura 5.5. Este arquivo externo pode ser gerado por qualquer editor de texto.

Capítulo 6 – Conclusão e trabalhos futuros.

6.1 Reafirmação da Contribuição

A carência de ferramentas computacionais específicas para o ensino de perícias em circuitos elétricos tem sido amenizada através de simuladores profissionais-comerciais ou através de softwares de análise matemática. Com o ambiente proposto, materiais instrucionais para o desenvolvimento de perícias na solução de problemas de circuitos elétricos-eletrônicos podem ser confeccionados e disponibilizados aos aprendizes em maior quantidade e com qualidade razoável.

Estes materiais possuem uma forte capacidade de simulação e interação com os estudantes, possibilitando assim que intrutores e aprendizes possam elaborar hipóteses e testá-las em um laboratório virtual. O ambiente permite, também o desenvolvimento de perícias segundo a teoria clássica de conceitos e de protótipos (Bussmann, 1991 e Lesgold, 1984).

Destaca-se ainda a necessidade do autor elaborar classes de topologias e problemas sobre circuitos específicos de acordo com os objetivos de ensino. Nesta fase, os professores mais experientes podem auxiliar os menos experientes, para obter circuitos elétricos-eletrônicos com finalidades instrucionais. Com isto, espera-se fomentar discussões contínuas entre os docentes das disciplinas de circuitos elétricos e eletrônicos.

Destaque deve ser dado também à forma de implementação em ambiente Cliente-Servidor em uma rede TCP/IP, possibilitando assim uma atualização “on-line” de exercícios que estarão disponíveis ao aprendiz. O

aprendiz e o professor-autor necessitam apenas executar um pequeno aplicativo em seu computador para que todo o armazenamento, recursos de simulação e geração de explicações fique disponível a partir de outro equipamento remoto.

Destacamos ainda, que a pouca diversidade de classes de gabaritos e meta-classes de componentes limitam o emprego deste ambiente como recurso pedagógico em sala de aula.

6.2 Trabalhos Futuros.

Como trabalhos futuros, são apresentados os seguintes pontos:

(a) a ampliação e implementação de novos gabaritos fornecidos pelo ambiente para a autoria dos exercícios, permitindo assim que novas abordagens pedagógicas possam se empregadas no desenvolvimento das perícias;

(b) a expansão da linguagem de autoria para permitir ao autor na ferramenta apresentada no Capítulo 5, a capacidade de criar novas classes de *gabaritos*, a ferramenta não deve estar limitada apenas aos recursos pedagógicos fornecidos pelo ambiente, a inclusão de novos recursos por parte do professor-autor permitirá uma troca de experiências entre docentes bem como abrir as portas para um estudo mais aprofundado quanto a eficácia de tal gabarito em relação a outro;

(c) a extensão dos conceitos e linguagem de autoria de tal forma a permitir a criação de novas meta-classes de componentes isolados - o professor-autor deve ser capaz de modelar o comportamento de novas meta-classes de elementos de circuitos elétricos, enriquecendo o ambiente

com recursos de análise ainda não disponíveis no ambiente atual. Como exemplo de meta-classe, poderíamos destacar a de resistores que variam o seu valor em função da temperatura;

(d) a gravação das ações do estudante de tal forma a tornar possível implementar um modelo de estudante. Através deste registro, um acompanhamento mais eficaz poderá ser feito do aprendiz, disponibilizando o modelo aos professores. Destaca-se ainda que este recurso abrirá as portas de uma outra linha de pesquisa que é o da seleção e sequenciamento dos exercícios a serem apresentados ao aprendiz, permitindo assim um aprendizado individualizado e tipificado de acordo com o seu modelo.

Referências Bibliográficas.

- ANDERSON, J.R.(1982), Acquisition of cognitive skills. *Psychological Review*, 1982, 89, 369-406.
- BLESSING, S.B.(1997), A programming by Demonstration Authoring Tool for Model-Tracing Tutors, *International Journal of Artificial Intelligence in Education*, volume 8, pp 233-261.
- BUSSMANN, S. (1991), Prototypical Concept Formation – Na Alternative Approach to Knowledge Representation, Technical memo TM-91-15.
- BROWN, J.S.; BURTON, R.R.; DE KLEER, J. (1982), Pedagogical, natural language and knowledge engineering techniques in Sophie I, II e III, In D. Sleeman & J.S. Brown (eds.), *Intelligent Tutoring Systems*, London Academic Press, pp 227-283.
- DIRENE, A.I. (1997), Designing Intelligent Systems for Teaching Visual Concepts. *International Journal of Artificial Intelligence in Education*, volume 8, pp 44-70.
- DIRENE, A.I. (1997b), Intelligent Training Shells for Operation of Digital Telephony Stations. *Anais da World Conference on Artificial Intelligence and Education (AIED'97)*. Kobe-Japan, pp 71-78
- DU BOULAY, B. e MIZOGUCHI, R. (1997), The Simquest Authoring System for Simulation-Based Discovery Learning, *Artificial Intelligence in Education*, IOS Press, pp 79-86.
- HOLLAN, J.M.; HUTCHINS, E.L; e WEITZMAN, L. (1984), STEAMER: na interactive inspectable simulation-based training system. *AI Magazine*, vol. 5, no. 2, pp. 15-27.

- HOWARD, R.W. (1987), Concepts and Schema Data: an Introduction, Cassel Education, 1987.
- JACKSON (1976), Introduction to Electric Circuits, fourth edition, Prentice-Hall, Inc. Englewood Cliffs, New Jersey.
- LYRA, A.C. (1999), O aprendizado de circuitos elétricos a partir de problemas exemplos utilizando simuladores. Congresso Brasileiro de Ensino de Engenharia (COBEENGE, 99).
- LESGOLD, A. M. (1984). Acquiring expertise. In Anderson, J. R. and Kosslyn, S. M., editors, Tutorials in Learning and Memory: Essays in Honor of Gordon Bower. W. H. Freeman.
- LESGOLD, A. M.; BONAR, J.G.; IVILL, J.M.; e BOWEN, A. (1987) , Na intelligent tutoring system for electronics troubleshooting: DC-circuit understanding. Technical Report. Learning Research and Development Center, University of Pittsburg, Pittsburg, Pennsylvania.
- MAJOR, N. (1991), Using COCA to build an intelligent tutoring system in simple algebra. Intelligent Tutoring Media, volume 2, No. 3/4, pp 159-169.
- MAJOR, N. (1997), REDEEM: Exploring Symbiosis Between Psychology and Authoring Environments. International Journal of Artificial Intelligence in Education, volume 8, pp 317-340.
- MERRYL, M.D.; TENNYSON R.D., Concept Classification and Classification Errors as a Function of Relationships Between Examples and Nonexamples, Improving Human Performance Quarterly, 1978, 7, 4, pp 351-364.
- MURRAY, T. (1998), Authoring Knowledge Based Tutors: Tools for Content, Instructional Strategy, Student Model, and Interface Design. Journal of the Learning Sciences, volume 7, número 1, pp 5-64.
- MURRAY, T. (1999), Authoring Intelligent Tutoring Systems : An analysis of the

- state of the art. *International Journal of Artificial Intelligence in Education*, 10, pp 98-129.
- NICHOLSON R.I.; SCOTT P.J. (1986); Computers and Education: the software production problem, *British Journal of Education Technology*, 1, 17, january 1986, pp 26-35.
- O'SHEA T. AND R.BORNAT AND B. DU BOULAY AND M.EISENSTADT AND I.PAGE (1984), "Tools for Creating Intelligent Computer Tutors", A. Elithorn and R.Barneji, *Human and Artificial Intelligence*, North-Holland, 1984.
- ROSCH, E. (1978), *Principles of Categorization, Cognition and Categorization*, Lawrence Erlbaum, 1978.
- SHARPLES, M. (1991), Computer-Based tutoring of Visual Concepts: from novice to expert, *Journal of Computer Assisted Learning* (1991), 7, pp 123-132.
- SHUTE, V.; e BONAR, J.G. (1986), An intelligent tutoring system for scientific inquiry skills. *Proceedings of the Eighth Cognitive Science Society Conference*, Amherst, Massachusetts, pp. 353-370.
- SHUTE, V.; e GLASSER, R. (1986), An intelligent tutoring system for exploring principles of economics. *Teaching Report. Learning Research and development Center*, University of Pittsburg, Pittsburg, Pennsylvania.
- SILVEIRA R.; VICCARI, R. (1997), Projeto Eletrotutor: Desenvolvimento e avaliação de ambientes inteligentes de ensino-aprendizagem, *Simpósio Brasileiro de Informática em Educação – 1997*, São José dos Campos, Brasil.
- TVERSKY, A. (1977), *Psychological Review*, 84, pp 321-352.
- TENNYSON, R.D.; PARK O.C. (1980), The teaching of concepts a review of instructional design research literature, *Review of Educational Research*,

Spring 1980, 50, 1, pp 55-70.

WHITE, B.Y.; FREDERIKSEN, J.R. (1985), Quest: qualitative understanding of electrical system troubleshooting, ACM SIGART Newsletter, no. 93, pp. 34-37.

WOOLF, B. (1987), Multiple Knowledge Sources in Intelligent Teaching Systems, IEEE Exeprt, Summer 1987, pp 41.